AD-A119 619 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 9/2
INTEGRATION OF FINITE ELEMENT ANALYSIS PROGRAM FOR CONDUCTION H--ETC(U)
JUN 82 W L ROBERTS

UNCLASSIFIED NL

1 OF 1

ADA
119619

END
DATE
FILMED
11-82
DTIC

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

INTEGRATION OF FINITE ELEMENT ANALYSIS PROGRAM FOR CONDUCTION HEAT TRANSFER WITH COMPUTER ANALYSIS LANGUAGE

by

Warren Leigh Roberts

June 1982

Thesis Advisor: G. Cantin

Approved for public release, distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A119 619	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Integration of Finite Element Analysis Program for Conduction Heat Transfer with Computer Analysis Language		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis; June 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Warren Leigh Roberts		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 91
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Finite Element
Interactive
CAL
FEAP
Conduction
Heat Transfer

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The Finite Element Analysis Program (FEAP) was modified and integrated with the Naval Postgraduate School version of the Computer Analysis Language (CAL-NPS). This enables the solution of linear and non-linear, two and three dimensional heat conduction problems in an interactive mode. The usual types of boundary conditions, including radiation, may be specified. The heat conduction group includes prompts for

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

user supplied data. Several existing CAL-NPS commands were improved and a "HELP" facility was added. Commands were added for visual display of the finite element mesh at graphics terminals. The User Guide for this expanded version of CAL-NPS is provided.

Approved for public release, distribution unlimited.

Integration of Finite Element Analysis
Program for Conduction Heat Transfer
with Computer Analysis Language

by

Warren Leigh Roberts
Lieutenant Commander, United States Navy
B.A., Macalaster College, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MECHANICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1982

Author: Warren Leigh Roberts

Approved by: Gilles Cantin
Thesis Advisor

[illegible]
Second Reader

P.J. Marto
Chairman, Department of Mechanical Engineering

William M. Tolles
Dean of Science and Engineering

ABSTRACT

The Finite Element Analysis Program (FEAP) was modified and integrated with the Naval Postgraduate School version of the Computer Analysis Language (CAL-NPS). This enables the solution of linear and non-linear, two and three dimensional heat conduction problems in an interactive mode. The usual types of boundary conditions, including radiation, may be specified. The heat conduction group includes prompts for user supplied data. Several existing CAL-NPS commands were improved and a "HELP" facility was added. Commands were added for visual display of the finite element mesh at graphics terminals. The User Guide for this expanded version of CAL-NPS is provided.

TABLE OF CONTENTS

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Professor Gilles Cantin for his invaluable guidance, help and friendship as instructor and thesis advisor. In addition, the author wishes to thank Professor Y. S. Shin for his interest, comments and thoughtful advice.

This thesis could not have been undertaken without the work by LCDR L. B. Elliot, USN, and LT J. M. Bettencourt, Portugese Navy, who prepared the way by installing CAL and FEAP, respectively, at Naval Postgraduate School.

Finally the author wishes to thank his wife, Mary, for her patience, understanding and encouragement throughout the course of this work.

I. INTRODUCTION

A. GENERAL DESCRIPTION

Since the implementation of the Finite Element Analysis Program (hereafter refered to as FEAP) at the Naval Postgraduate School, it has had only limited use. It is now available to general NPS users in an interactive form through the IBM 3033 VM/MVS time sharing system. The data management and user interactive command structures were established within the existing interactive program, Computer Analysis Language (CAL). This integration of systems provides the ability to solve linear and nonlinear, steady and unsteady, two and three dimensional heat conduction problems involving temperature dependent thermophysical properties and complicated radiation/convection boundary conditions.

Additional capability was provided to CAL system users both in changes to existing subroutine groups and in the addition of a new graphics group with its attendant command structure. The graphics functions enable the user to plot two- and three- dimensional structural and heat transfer meshes. Another new facility is the HELP operation which allows a user experiencing trouble with a particular operation to interactively obtain assistance.

B. HISTORICAL BACKGROUND

The original CAL program was developed by Professor E. L. Wilson of the University of California in 1977 [Ref. 1]. It was later adapted and modified for use at the Naval Postgraduate School by LCDR L. B. Elliot [Ref. 2]. The FEAP program was written by Professor R. L. Taylor of the

University of California [Ref. 3] in 1977. Implementation of FEAP at Naval Postgraduate School was done by LT J. M. Bettencourt [Ref. 4].

C. OBJECTIVES

The objectives of the author's work have been to:

1. integrate the data management system of FEAP with CAL to create an interactive conduction heat transfer problem solving system;
2. modify existing operations to extend their usefulness;
3. add operations to extend the capabilities of the program to include graphics;
4. create a HELP facility;
5. create a USER'S MANUAL to facilitate use of this program.

II. ORGANIZATION OF HEAT TRANSFER GROUP

This chapter provides a general overview of the organization of the Heat Transfer Group of CAL. It is intended to provide sufficient information to permit users to operate the CAL heat transfer package.

The execution of the program is flexible and controlled by user selection of operations in a logical sequence from the commands that are available. There are two broad categories of operations, data input and problem solution.

A. DATA INPUT

1. Initialization

The heat transfer group performs matrix creation and manipulation automatically. As the problem progresses, arrays are created, altered and deleted under program control. Through the HTXFR operation the user provides sufficient information to establish the initial arrays for data input and problem solution. The number of nodes, number of elements, number of material sets, spatial dimension, number of degrees of freedom per node and the maximum number of nodes per element are required. For heat transfer problems the number of degrees of freedom per node is always one. The option to assign a higher number is available because the equation solvers in this program are applicable to other fields of which future work may make use.

2. Nodal Coordinates

Nodal coordinates are input via the COORD operation. This operation has built in node generation capability. By specifying an initial point and a node generation vector,

the user may easily input large meshes. Coordinate system conversion is also available. Coordinates may be input in the Cartesian system, the cylindrical system with any one of the three axes longitudinal, the spherical system or any combination of the above systems. All coordinates are converted to Cartesian coordinates for use with CAL.

3. Element Connectivity

The ELCON operation inputs the element connectivity data. Here again is a generation capability. The user may specify the connectivity for one element and a generation vector to create additional rows.

For the two-dimensional elements the user may specify a 4 to 9 node isoparametric element. There is an 8 to 21 node isoparametric element for three-dimensional elements. Both of these elements must follow the numbering convention shown in Appendix A.

4. Material Properties

The required amount of material property information varies from problem to problem. The PROP operation prompts the user for the information required to solve the problem at hand. At a minimum the material's conductivity (k), specific heat (c), specific mass (rho),heat generation per unit volume (q''') and the geometry type (plane or axisymmetric) must be specified in a consistent system of units. Appendix A includes examples of consistent systems.

The user also inputs the number of Gaussian points per direction for quadrature and codes for temperature dependent properties and boundary conditions. The codes indicate other information required. Temperature dependent properties are input as tables and linear interpolation is used to determine the property value at a given temperature. Boundary conditions are specified as shown in Appendix A.

If an exterior boundary line or surface condition is not specified, it is assumed to be insulated.

5. Constant Temperature Nodes

For problems requiring certain nodes to be at constant temperature, the CTEMP operation is available. This operation may also be used during the solution stage of the problem to provide step changes at previously specified constant temperature nodes. Because this operation generates an array used in profiling the solution equations, the user may not change the node numbers that were established as constant constant temperature nodes after execution of the PROF operation. The temperatures of these nodes, however, may be changed.

6. Equation Profile

The PROF operation establishes the equation profile for problem solution. Prior to the execution of this operation, any data input may be changed by specifying the appropriate operation and re-entering the data. After its execution the user may not change the nodes designated as constant temperature nodes to temperature varying nodes or vice versa. The user may change the value of the constant temperatures.

B. SOLUTION

The matrix formulation of the heat transfer problem as discussed in Reference 4 is:

$$[K]\{T\} + [C]\{\dot{T}\} + \{F\} = \{0\} \qquad (1)$$

where (K) represents the conductivity matrix, (C) represents the heat capacity matrix, {F} represents the flux vector and {T} represents the temperature vector. The derivative of {T} with respect to time is $\{\dot{T}\}$. The flux vector includes

heat generated per unit volume and boundary fluxes as specified for given boundary surfaces. This is a fully generalized formulation, including non-linearities, since the matrices [K] and [C] and the vector {F} can be temperature dependent.

1. Forming Conductivity Matrix

For heat transfer problems, which involve only one degree of freedom per node, the conductivity matrix, [K], will always be symmetric. The capability for generating unsymmetric matrices was provided, but will only be applicable when additional types of problems are programmed into the CAL system. The command for the unsymmetric conductivity matrix formulation is USYMC.

2. Forming Heat Capacity Matrix

The heat capacity matrix, [C], used in time dependent problems, can be generated with either CCAP or LCAP. To form a consistent capacitance approximation use the operation CCAP. A lumped capacitance approximation is formed using the LCAP operation.

3. Forming Flux Vector

To complete the problem formulation, the flux vector, {F}, must be generated. The FORM operation forms the flux vector taking into account the internal heat generation and boundary surface fluxes as indicated in the PROP operation.

4. Equation Solving

Once the time independent problem is formulated, the temperature vector, {T}, is calculated by the CALC operation. Time dependent problems do not use this operation, but rather the ordinary differential equation solver.

5. First Order Ordinary Differential Equation Solver

The first order ordinary differential equation solver is accessed with the ODE operator. It employs the Zienkiewicz two- and three-level schemes [Ref. 4, 5].

In addition to the time step size change using the DTIM operation, an optional automatic time step adjustment is incorporated in the ODE operation. The norm of the difference between temperature vectors at two consecutive times is computed at each step. If the norm is less than a user specified maximum temperature difference, the time step is doubled before going to the next step. If the norm is greater than a user supplied minimum temperature difference, the time step is halved and calculation for that time step is repeated until the norm is acceptable. If the temperature differences are specified as zero, no time step adjustment will be performed.

The user specifies one of three functions which are performed by the ODE operation. A second operation name, which must be separated by a comma, follows ODE. The options are INIT, LINE or QUAD.

The operation ODE,INIT is used to input the integration constants theta, beta and gamma [Ref. 5], the maximum and minimum temperature differences for the automatic time step adjustment and the initial temperature vector. No time integration is performed by this instruction.

ODE,LINE performs the two point scheme and the current temperature vector is substituted by the newly calculated temperature vector.

The ODE,QUAD operation is similar to the ODE,LINE operation but uses the three point scheme.

6. Printing Nodal Temperatures

Once the temperature vector has been updated, the PTEMP operation prints the temperatures in node number order.

III. ORGANIZATION OF GRAPHICS GROUP

The graphics group is capable of displaying meshes either on the IBM 3277 dual screen terminal system or any PLOT-10 compatible terminal. It is initiated through the use of the GRAPH operation by which the user specifies the type of graphics terminal in use.

A. TITLE

The TITLE operation is used to title the mesh being displayed. The user may input up to three lines of fifteen characters for the title. This operation must be specified immediately prior to the displaying operation.

This operation calls the USRIN subroutine which reads from the terminal three lines of characters. A flag (IFLAG) is set to indicate a title is to be printed.

B. HEAT TRANSFER MESH

The PLHX operation locates the heat transfer coordinate and connectivity arrays and displays the mesh. The viewing area is optimized so the longest dimension is full screen. The maximum and minimum coordinate values for each direction are displayed. This operation displays two- and three-dimensional meshes. If the mesh is three-dimensional, the user must specify the viewing plane.

This operation calls either the FPPLOT (2-D) or FP3PLT (3-D) subroutine. These subroutines scan the coordinate arrays for maximum and minimum values and initialize the graphics screens. They both use subroutine BOX to set the virtual window, set the screen window and draw a box around the plotting area. The standard element connectivity is

stored in an array which is used in conjunction with the user's element connectivity matrix to draw the elements line by line. The center node is plotted using a "+" symbol. The user's title is plotted (if IFLAG is not 0) by subroutine USRTIT, which then resets IFLAG. The maximum and minimum values of the vertical and horizontal coordinates are printed by SCRDAT. Subroutine SCRDAT also indicates axis orientation. Prior to terminating the screen graphics, subroutine TITLE is used to write the title box identifying the type of mesh being displayed.

C. STRUCTURAL MESH

The PLST operation displays the structural system specified by user supplied coordinate and connectivity arrays. The viewing area is optimized as described for the PLHX operation. Likewise, the user may specify the viewing plane as either the X-Y, Y-Z or X-Z plane.

This operation calls the CLPLOT subroutine. CLPLOT uses the same logic and subroutines as PPPLOT. The difference between the two subroutines is in the method of storage of the coordinate arrays and the connectivity matrix.

IV. CHANGES TO EXISTING GROUPS

Modifications were made in several existing subroutine groups to make their operations more versatile. One utility subroutine was improved and one was added.

The changes that were made are sensitive to prior versions of CAL. The same results are obtained for previously existing operation commands. No files used with other editions of CAL need to be modified to operate with this program.

A. UTILITY SUBROUTINES

1. Subroutine RCARD

Subroutine RCARD reads and interprets the operation commands. An operation command has the form

OP,M1,M2,M3,M4,M5,N1,N2,N3,N4

where OP is the operation name, M1 to M5 are matrix names and N1 to N4 are integers. Previously, N1 to N4 had to be values greater than or equal to zero. The symbol/state logic matrix and subsequent action codes were modified to allow users to input negative integer values.

This is important for the ZERO operation which enables users to create matrices with a given value in the diagonal locations and another value in the off diagonal locations. Prior to this change, a user desiring to create such a matrix with negative values had to input the matrix row by row.

2. Subroutine FRTCMX

Subroutine FRTCMX is a new utility subroutine which allows the CAL program to invoke most CP/CMS commands.

After the invoked command is executed, control returns to CAL.

It is presently used in the SAVE and RESUME operations to invoke the CMS command

FILEDEF NSAVE DISK M1 SAVE (RECFM VS LRECL 7290 BLKSIZE 7294)

where NSAVE is a logical unit number assigned to the SAVE and RESUME operations and M1 is a user input file name.

B. GROUP 1

1. SAVE Operation

The save operation creates a file on the user's A-disk containing all arrays in storage at the time of issuance. Previously the entire 100,000 word main array was stored, regardless of how many words were actually being used. It was always stored under the name FILE 02, preventing the user from saving more than one problem.

The method of storing the array was changed to store just the number of locations actually being used. More than one problem may be saved because the subroutine FRTCMX was used to create a SAVE file with a name assigned by the user. If a name is not specified FILE 02 will be the name of the saved problem.

2. RESUME Operation

The RESUME operation reads a saved file into memory. It was altered to read named files saved by the new SAVE operation. If a name is not specified FILE 02 will be read.

3. LOADI Operation

The LOADI operation loads integer arrays. The arays were input row by row. The option to generate arrays was added. The user may specify one row and a row generation

vector. The number of rows specified will be automatically generated. This operation was moved from the static analysis subroutine group to the general matrix command group.

4. PRINT Operation

The PRINT operation prints an array in matrix format. Previously it could only print arrays containing real numbers, resulting in the erroneous printing of arrays containing integers. The user may now specifiy whether the array to be printed contains real or integer values.

5. HELP Operation

The HELP operation was added to the general command group and provides the user with information on the use of all the available operations. It accesses a file of instructions, sorts through them to find the desired operation and displays the appropriate information on the screen.

C. GROUP 2

1. NODES Operation

The NODES operation creates the matrix of nodal coordinates for a structural problem. This information was entered in cartesian coordinates, node by node.

The user may now opt to enter data in cartesian, cylindrical (any axis longitudinal) or spherical coordinates. It will be converted to the cartesian coordinate system used by CAL. Additionally, the user may generate new nodes by specifying one node and a node generation vector.

V. SOLUTION OF CONDUCTION HEAT TRANSFER PROBLEMS

In order to solve a conduction heat transfer problem, the user must provide the solution algorithm to CAL. A discussion of the matrix manipulation and equation solving techniques can be found in Reference 4 . This chapter will present possible algorithms for solution of this class of problems.

A. STEADY STATE PROBLEMS

These problems take the form:

$$[K]\{T\} + \{F\} = \{0\} \tag{2}$$

1. Linear Heat Conduction Problems

This is the simplest case to consider. After inputting mesh data, the user must form the conductivity matrix [K], (SYMC) and the load vector {F}, (FORM). Then the nodal temperatures must be calculated (CALC) and printed (PTEMP).

Consequently the sequence of solution operations for this type of problem would be:

```
SYMC
FORM
CALC
PTEMP
```

2. Non-linear Heat Conduction Problems

Since the conductivity matrix is time dependent on temperature, an iterative algorithm must be used. This requires a looping operation (LOOP, NEXT) around the linear steady state sequence.

The solution operations may be:

LOOP,N1
SYNC
FORM
CALC
PTEMP
NEXT

where N1 is the user's guess of the number of iterations necessary to obtain equilibrium. However, the program maintains an internal check on the residuals. When they decrease below the predefined tolerance (TOL command, default is 10^{-9}), the looping operation is terminated upon the subsequent NEXT command.

B. TIME DEPENDENT PROBLEMS

These problems involve the full form of equation (1):

$$[K]\{T\} + [C]\{\dot{T}\} + \{F\} = \{0\} \tag{1}$$

1. Linear Heat Conduction Problems

This case requires the solution of a first order ordinary differential equation (ODE). Additionally a heat capacity matrix, [C], must be formed (CCAP or LCAP) and a time step provided (DTIM).

The differential equation solver is accessed using the operation ODE,N1 where N1 is one of the following:

INIT to specify initial temperature vector and the integration constants
LINE to perform the two-time level algorithm
QUAD to perform the three-time level algorithm

The heat capacity and conductivity matrices are unchanged in a linear problem and must be placed outside the loop. The load vector is reformed every time step and the

time must be advanced with ADTIM. These operations are included in the loop.

The sequence of operations to solve a linear time dependent problem may be:

DTIM
ODE,INIT
SYMC
CCAP (or LCAP)
LOOP,N1
FORM
ODE,LINE (or QUAD)
ADTIM
PTEMP
NEXT

where N1 is the number of time steps the user wants to take.

2. Non-linear Heat Conduction Problems

The heat capacity and/or conductivity matrices are temperature dependent in this class of problem. The temperature dependent matrix (matrices) must appear inside the looping operation whereas the constant property matrix would be excluded from the loop.

To solve a fully non-linear problem the following operation sequence may be used:

DTIM
ODE,INIT
LOOP,N1
SYMC
CCAP (or LCAP)
FORM
ODE,LINE (or QUAD)
ADTIM
PTEMP
NEXT

where N1 is the number of time steps the user wants to take.

C. NUMERICAL EXAMPLES

1. Hollow Cylinder with Circumferential Heating Strips

A hollow cylinder with a four inch outer diameter and a three inch inner diameter was subjected to an axial forced connvection condition in a wind tunnel by Professor P. F. Pucci of the Naval Postgraduate School. There were sixteen one-quarter inch wide heating strips equally spaced over 180 degrees of the outer surface as illustrated in Figure 1.

The heating strips were maintained at a constant temperature of 160° F and measurements of the surface temperatures between the strips were made using teledeltos paper. The ambient temperature was 60° F. The cylinder was considered to have a density of 70 lb/ft^3 and a specific heat of 0.6 BTU/lbm°F. Tests were made with four heat transfer coefficients (h).

The finite element model took advantage of the cylinder's symmetry, consisting of one half of a heating strip plus one half of the interval between strips. The element mesh,as generated by the PLHX operation, is shown in Figure 2. The comparison of the model data to the teledeltos paper measurements was very favorable and is shown in Figure 3.

2. Transient Surface Temperatures in an Infinite Plate

An infinite flat plate was considered as a test problem (Figure 4). The plate was at an initial uniform temperature greater than the ambient temperature and then exposed to convection conditions with a constant external heat transfer coefficient (h).

The approximate solution was obtained using a Heisler chart. The temperature of the outer wall was computed at 1 minute, 5 minutes and every 5 minutes there

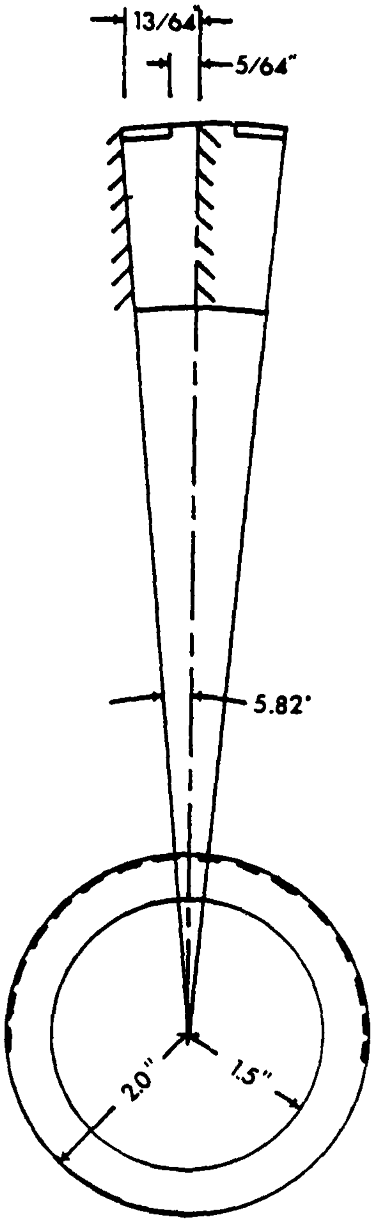
after up to 30 minutes. The comparison between these values and those generated by CAL was very close and is shown in Figure 5.

VI. CONCLUSIONS AND RECOMMENDATIONS

The code that was integrated into CAL provides an accurate and reliable means for solving a variety of conduction heat transfer problems. The system is user friendly both in prompting for input and detecting errors. The use of the heat transfer group of commands is encouraged, as well as efforts to increase its versatility.

While the capabilities of the program are significant, there is room for improvement. The present version uses primarily an in-core solution technique, which restricts the problem size to within the user's virtual machine space. The capacity to handle larger problems may be increased through the use of an out of core technique for building and storing the conductance and capaitance matrices, as well as equation solving.

The graphics package could be expanded and improved to provide more information to the user. The plotting of isoparametric elements needs to include the capability to generate and plot curved lines. Another desirable capability is to portray all three axes on the screen with apparent depth and allow rotation to any desired view. A capabilty to plot isotherms could be added to enhance the interpretation of the results.



Hollow Cylinder

Figure 1.

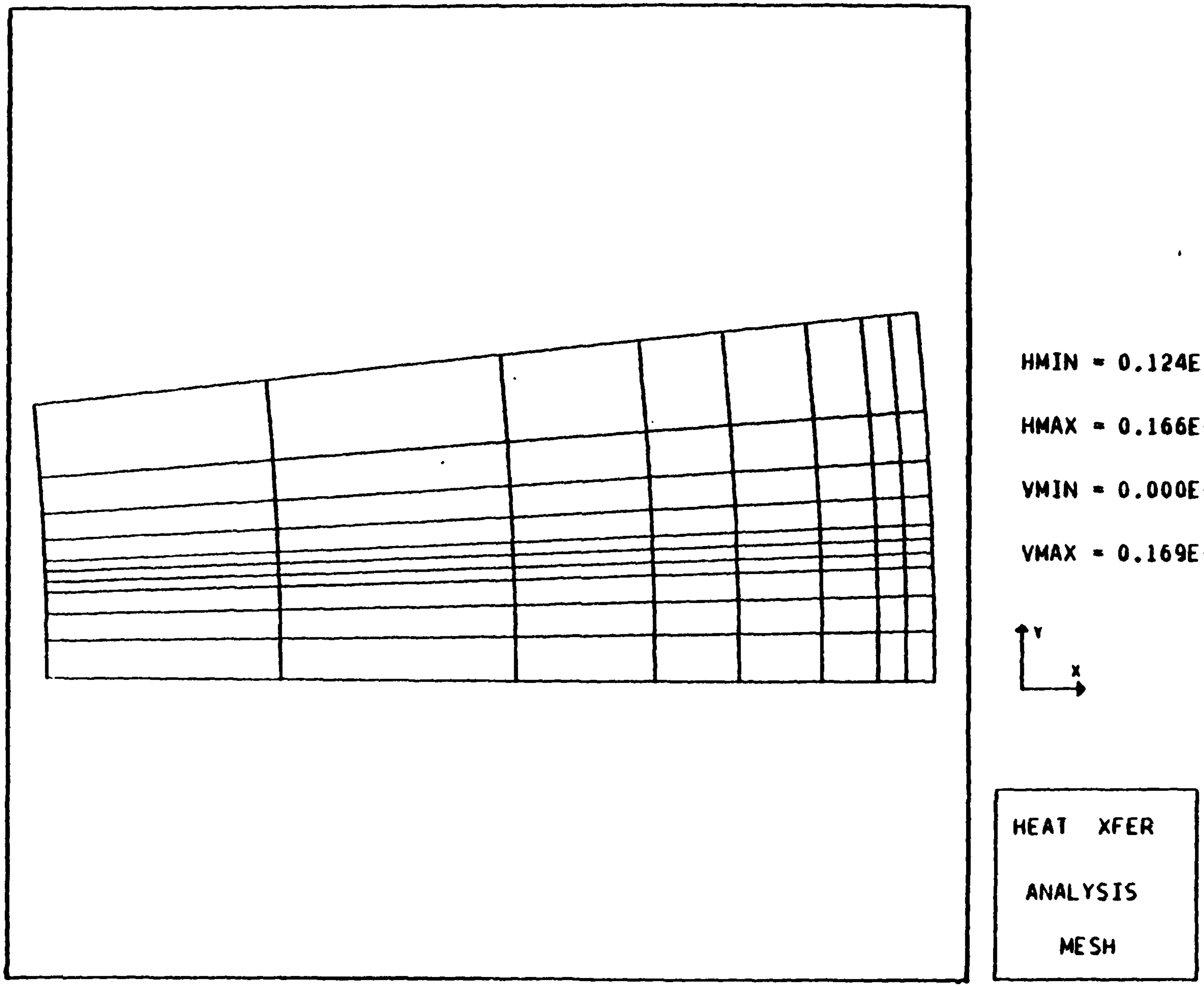
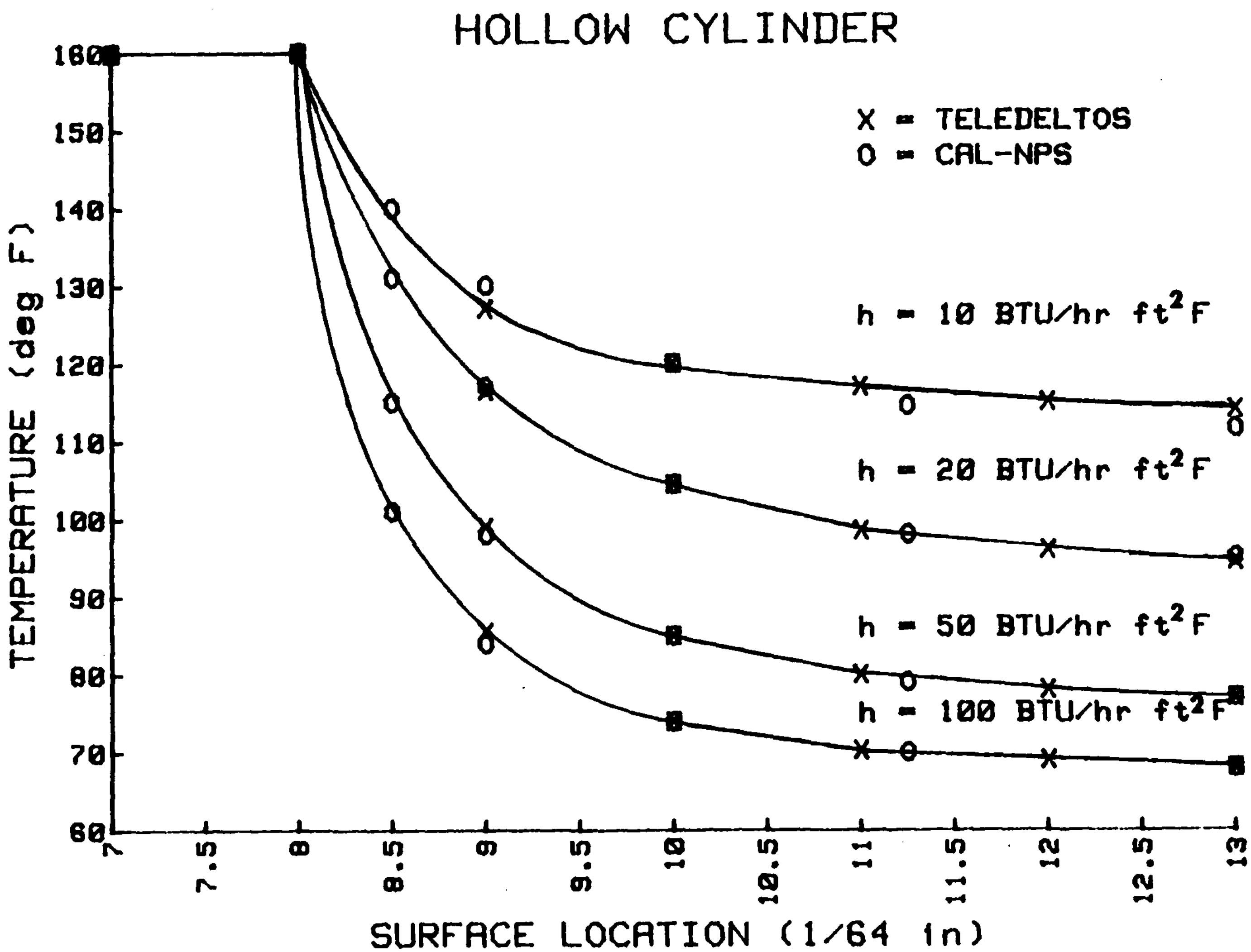
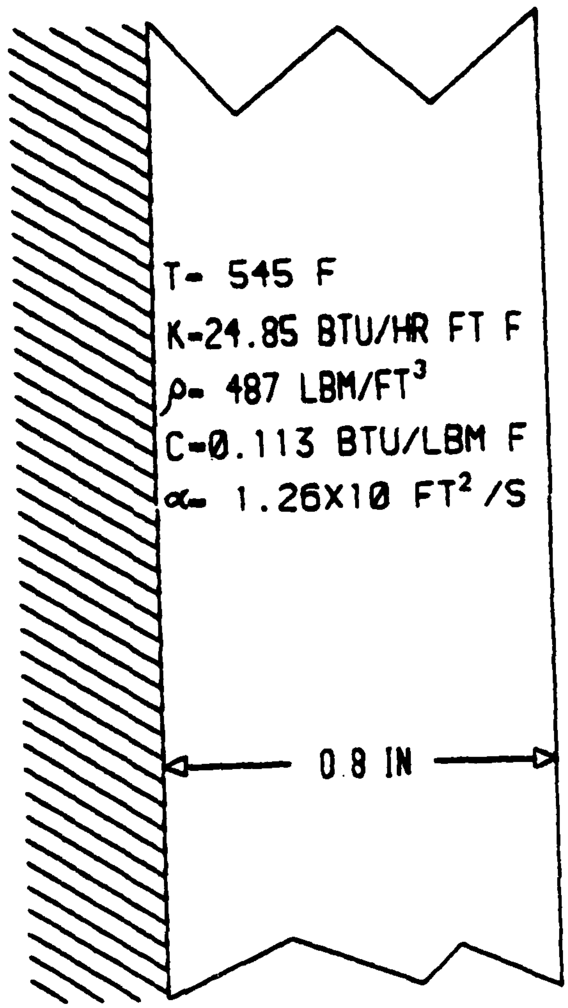


Figure 2. Graphic Display



HOLLOW CYLINDER
X = TELEDELTOS
O = CAL-NPS
h = 10 BTU/hr ft^{2}F
h = 20 BTU/hr ft^{2}F
h = 50 BTU/hr ft^{2}F
h = 100 BTU/hr ft^{2}F
TEMPERATURE (deg F)
160
150
140
130
120
110
100
90
80
70
60
7
7.5
8
8.5
9
9.5
10
10.5
11
11.5
12
12.5
13
SURFACE LOCATION (1/64 in)

Figure 3.



Infinite Slab

Figure 4.

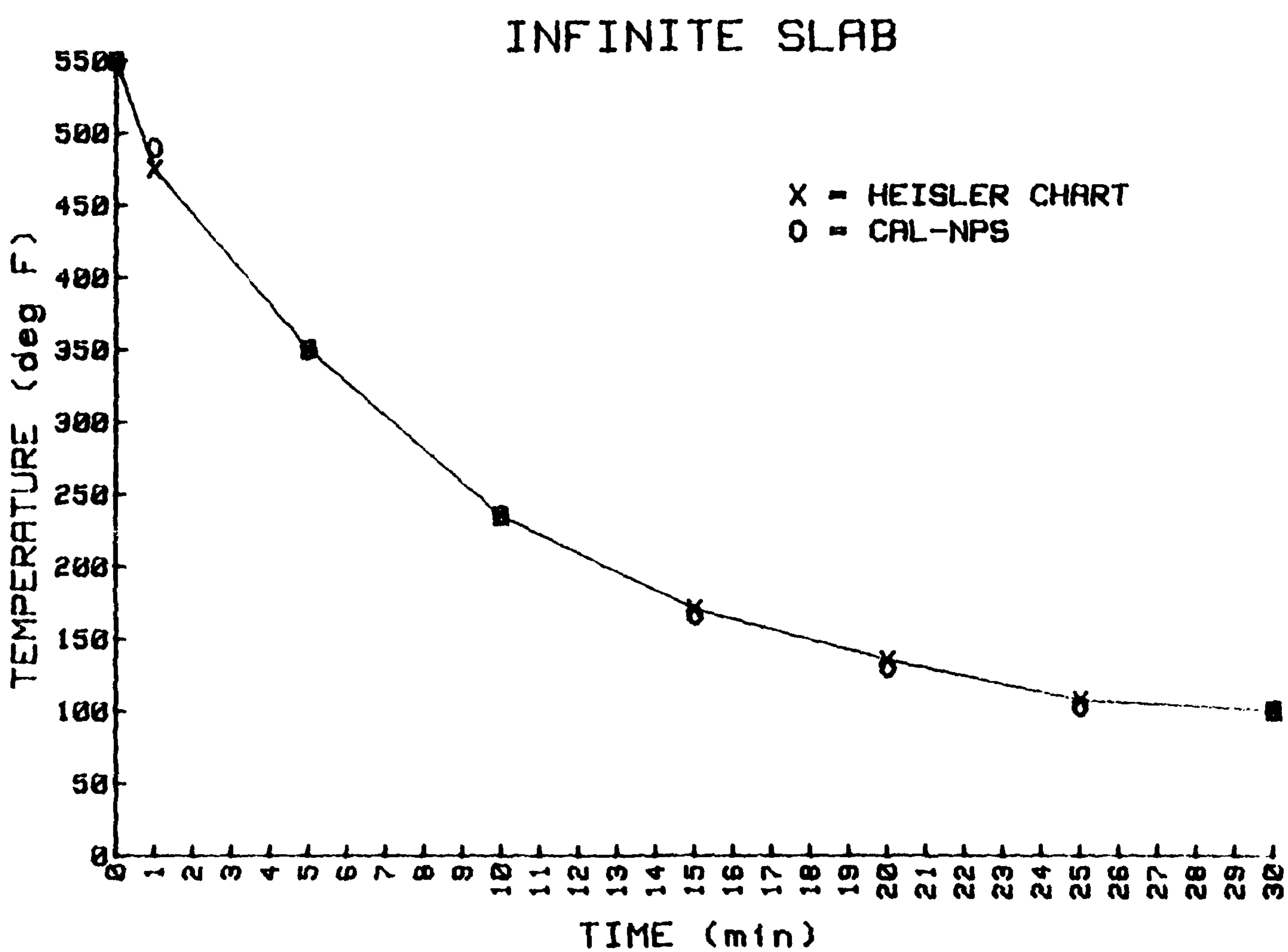

INFINITE SLAB
X = HEISLER CHART
O = CAL-NPS
TEMPERATURE (deg F)
0
50
100
150
200
250
300
350
400
450
500
550
TIME (min)
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30

Figure 5.

APPENDIX A

USER'S GUIDE

This appendix provides details on the use of CAL with the IBM 3033 computer at NPS. The program as modified at the Naval Postgraduate School is subsequently referred to as CAL-NPS. Section A provides the details on the command structure. Section B is a summary of commands available. Section C provides the job control language for executing the program in both the batch and interactive modes at NPS. Section D contains detailed specifications for each available command. Finally, section E gives direction for solving larger problems with CAL-NPS. The majority of this appendix was originally published as Reference 1 . The author wishes to express appreciation to Professor Wilson for permission to use this material.

A. FORM AND RESTRICTION OF THE LANGUAGE

CAL-NPS is an interpretive language which is designed to manipulate arrays and matrices, to perform standard structural analysis operations and to perform conduction heat transfer analysis operations. A CAL-NPS program run involves the reading of the input deck once and executing the commands designated by the operation cards as they are encountered. Looping operations allow a sequence of commands to be executed more than once.

The input deck is composed of operation cards and data cards. The data cards directly follow each operation card which requires data (see LOOP operation for exception to this). The operation card contains the name of the operation to be executed, names of arrays associated with the

operation and integer constants. Examples of the general form of this card are:

```
OP,M1,M2,M3,M4,M5,N1,N2,N3,N4  comments
OP,M1,N1,N2
OP,N1
OP
```

in which OP is the name of the operation to be executed, Mi is the name of an array and Ni is an integer. The names of OP or Mi are one to eight alphabetic or numeric characters to be selected by the user. The first character of a name must be alphabetic. The sequence of terms OP, Mi and Ni must be separated by commas. Characters following a blank will be printed as comments in the output from the program run.

If an operation attempts to load or generate an array which previously existed, the program will delete the array before the execution of the operation. A new array need not be the same size of the old array which had the same name.

B. SUMMARY OF COMMANDS

1. General Commands

* indicates a significant change or addtion in CAL-NPS

START - Initialize for the next problem
STOP - Normal termination
NO - Temporary suppression of output
YES - Restores output
LABEL - Print comments
READ - Change logical device for input
WRITE - Change logical device for output
TIME - Suppress time printout
SAVE - Interrupt a problem *
RESUME - Continue an interrupted problem *
LIST - List arrays and storage used
HELP - Access HELP files

2. General Matrix Commands

LOAD - Load user defined real matrix
LOADI - Load user defined integer matrix
ZERO - Create null or unit matrix
PRINT - Matrix print operation *
DUP - Matrix duplication
ADD - Matrix addition
SUB - Matrix subtraction
MULT - Matrix multiplication
TRAN - Matrix transpose
SCALE - Multiply a matrix by a scalar
SOLVE - Solution of linear equations
DUPSM - Form sub-matrix from large matrix
STOSM - Store sub-matrix in large matrix
DUPDG - Form row matrix from diagonal
STODG - Store row on diagonal

MAX - Evaluate row maximums

NORM - Evaluate matrix norms

INVEL - Invert each term in matrix

SQREL - Square root of each term in matrix

LOG - Natural log of each term in matrix

PROD - Evaluate product of all terms in a matrix

DELETE - Delete matrix from storage

3. Static Analysis Operations

NODES - Input structural joint geometry *

BOUND - Specify boundary conditions

BEAM - Form 3-D beam stiffness matrix

TRUSS - Form 3-D truss stiffness matrix

PLANE - Form 3 to 8 node plane stiffness matrix

SLOPE - Form stiffness matrix from slope/deflection eq.

FRAME - Form 2-D frame stiffness matrix

LOADS - Form load vector

ADDSF - Form global stiffness and mass matrices

ADDK - Add element matrix to global matrix

MEMFRC - Calculate element forces from joint displacements

DISPL - Print joint displacements

FORCE - Evaluate and print member forces

4. Dynamic Analysis Operations

FUNG - Generate equal interval time function

STEP - Integrate dynamic equilibrium equations

EIGEN - Evaluate mode shapes and frequencies

DYNAM - Evaluate uncoupled equations of motion by mode superposition method

PLOT - Line printer plot of joint time history

5. Heat Transfer Operations

HTXFR - Initiate heat transfer problem *

COORD - Input nodal coordinates *

ELCON - Input element connectivity matrix *

PROP - Input material property data *

CTEMP - Input constant temperature node data *

PROF - Establish profile of equations *

SYMC - Create symmetric conductance matrix *

USYMC - Create unsymmetric conductance matrix *

LCAP - Create lumped capacitance matrix *

CCAP - Create consistent capacitance matrix *

FORM - Create flux vector *

CALC - Solve time independent systems of equations *

ODE - Solve time dependent systems of equations *

PTEMP - Print nodal temperatures *

TOL - Set solution convergence tolerance *

CONV - Perform temperature convergence test *

DTIM - Set time step increment *

ADTIM - Advance time by one time step *

PROMPT - Suppress/restore prompts *

6. Graphics Operations

GRAPH - Initiate graphics *

TITLE - Label mesh plot *

PLHX - Plot heat transfer mesh *

PLST - Plot structural mesh *

7. Loop Operations

LOOP - Start of loop

NEXT - End of loop

SKIP - Conditional skip of operations within loop

8. Names Available for User Subroutines

USERA

USERB

C. EXECUTION OF CAL-NPS

For the time sharing (CMS) system at Naval Postgraduate School, do the following:

```
_(Use the standard LOGON procedue)
_link 0040P 191 199
ENTER PASSWORD:
_XXXXX        (ESAN)
R; T=0.01/0.01 11:09:55
_access 199 C            (Note: You must access the "C"
C (199) R/O               disk)
R; T=0.01/0.01 11:10:04
_cal
        ENTER TERMINAL CODE:

     1 = PLOT-10 Compatible Terminal (GRAPHICS)
     2 = IBM 3277 DUAL SCREEN (GRAPHICS)
     3 = Any Alpha Numeric Terminal (NO GRAPHICS)

_1 (or 2 or 3, as appropriate)
(The computer responds with several lines of procedure)
EXECUTION BEGINS ......
------------------------------------------0.0 SECONDS
(You are now under the control of CAL-NPS)
start
**START
(Your own CAL-NPS program is inserted here)
stop
**STOP
R; T=0.01/0.01 11:12:45
_log (Terminates session)
```

D. CAL-NPS COMMAND SPECIFICATIONS

1. General Matrix Operations

CAL-NPS has most of the standard matrix operations plus some special array operations which are useful in engineering analysis. The following is a list of approximately 32 operations which are used for control and general matrix manipulation.

A "+" indicates the formation of a new matrix. A matrix previously defined with the same name will be deleted. A "-" indicates modification of an existing matrix.

Note: Whenever the expression "card" is used it is meant to also stand for "instruction" in interactive mode.

START

This operation eliminates all arrays which were previously loaded or generated.

STOP

This operation causes normal termination of a CAL-NPS program.

NO
YES

These operations are used to selectively suppress output from CAL-NPS. The NO operation suppresses all printing, except diagnostics, until the operation YES is encountered. Therefore, in subsequent runs of the same CAL-NPS program, output which was previously correct need not be reprinted if these cards are inserted in the data deck.

LABEL,N1

This operation will read and print N1 comment cards which follow the operation card. Column 1 of each card will be interpreted as a standard carriage control symbol (i.e. 0 for double space and 1 for skip to the top of the next page).

READ,N1

THIS OPERATION IS VALID ONLY WITH THE CP/CMS TIME SHARING SYSTEM.
This operation permits the selection of a user's file or the terminal as the input file device. The default is the terminal. If N1 is 4, subsequent commands will be read from FILE FT04F001 on the user's A-disk. If N1 is 5, the terminal will be restored as the input file device. All disk files prepared for use with this command should end with either STOP or READ,5. This command will not be executed on the MVS (batch) system.

WRITE,N1

THIS OPERATION IS VALID ONLY WITH THE CP/CMS TIME SHARING SYSTEM.
This operation permits the selection of a file on the user's A-disk or the terminal as the output file device. The default is the terminal and all error messages will be printed at the terminal regardless of the output file device selected. If N1 is 8, subsequent ouput will be written into FILE FT08F001 on the user's A-disk. If N1 is 6, the terminal will be restored as the output file device. This command will not be executed on the MVS (batch) system.

TIME

This operation permits the time printout to be suppressed without loss of other output. A second TIME will restore the time printout unless the print output is suppressed with the NO command.

SAVE or SAVE,M1 or SAVE,M1,N1

This operation saves all arrays in storage at the time of issuance. Saved arrays will contain all modifications made since their creation. M1 and N1 are optional and if not included, the arrays will be stored in FILE 02 on the user's A-disk. The saved files will be assumed to be for general matrix manipulation or a structural problem. M1 is the file name (up to six letters) under which the user wishes to store the arrays. The file type will be SAVE. If N1 is 1, a general matrix manipulation or structural problem is being saved. If N1 is 2, a heat transfer problem is being saved.

RESUME or RESUME,M1 or RESUME,M1,N1

This operation reads a saved file into memory. Any arrays currently in storage will be destroyed. A file must have been previously created on the user's A-disk using the SAVE operation. M1 and N1 are optional and if not included, FILE 02 will be read. It will be assumed to be a general matrix manipulation or structural problem. M1 is the file name assigned to the saved file on the user's A-disk. The file type must be SAVE. If N1 is 1, a general matrix manipulation or structural problem is being resumed. If N1 is 2, a heat transfer problem is being resumed.

LIST
The LIST operation prints the directory information for arrays in storage and the amount of storage used.

LOAD,M1,N1,N2,N3

This operation will load an array of real numbers named M1 which has N1 rows and N2 columns. The terms of the array are punched in row-wise sequence on data cards following this operation. N3 is optional. If N3 is zero or blank, the cards are punched in (8F10.0) format. If N3 is one, an additional card which contains the format of the data cards must preceed the data. For example, if the data is to be 4 numbers per card in field widths of 15, the additional card would contain the following information: (4F15.0). If N3 is nine, the data cards will be read in free format.

LOADI,M1,N1,N2,N3,N4 or LOADI,M1,N1,N2,,N4 or
LOADI,M1,N1,N2,N3 or LOADI,M1,N1,N2 or
LOADI,M1,M2,N1,N2,N3,N4

This operation will load an integer array named M1 which has N1 rows and N2 columns. The terms of the array are punched in row wise sequence on data cards following this operation. M2,N3 and N4 are optional. If N3 is zero or blank, the data must be punched in (16I5) format. If N3 is one, an additional card containing the format of the data cards must follow this operation and preceed the data. For example, if the data is to be 4 numbers per card in field widths of 10, the additional card would contain: (4I10). If N3 is nine, the data will be read in free format.
If the letters "INCR" are placed in the position of M2, this operation has an increment generation capability. Data must be entered as follows:

Item	Contents
1	Row number
2	Value 1
3	Value 2
etc.	
N2+1	Value N2
N2+2	Generation code

If the generation code is not zero, the next card must contain the following:

Item	Contents
1	Row number increment
2	Value 1 increment
3	Value 2 increment
etc.	
N2+1	Value N2 increment
N2+2	Last row to be generated

ZERO,M1,N1,N2,N3,N4

A real matrix named M1 is created with N1 rows and N2 columns. The terms in this matrix will have the following values:

M1(I,I) = N3   I = 1,...N1
M1(I,J) = N4   J = 1,...N2

Therefore this operation can be used to form null or unit matrices.

PRINT,M1 or PRINT,M1,N1 or PRINT,M1,N1,N2 or PRINT,M1,N1,N2,N3

This operation will print the array named M1 in a matrix format of up to eight columns per line. N1,N2,N3, and N4 are optional. N1 is the number of comment cards (following the operation card) which will be read and printed. N1 defaults to zero. If N2 is included, the matrix will be printed in partitioned form with N2 columns per partition. Lines will have N2*15 + 5 characters. N2 defaults to 8, printing 125 charcters per line. If N3 is included and is greater than zero, integer format (I6) is used. The default value is zero and real format (PD15.7) is used. The user is cautioned not to overcome the capacity of the displaying device in use to avoid wrap around on the screen.

DUP,M1,$\overset{+}{M}$2

This operation will form an array named M2 which is identical to the array named M1.

ADD,$\overset{-}{M}$1,M2

This operation will replace matrix M1 with the sum of the matrices M1 and M2.

SUB,$\overset{-}{M}$1,M2

This operation will replace matrix M1 with matrix M1 less matrix M2.

MULT,M1,M2,$\overset{+}{M}$3

This operation generates a new matrix M3 which is the product of matrices M1 and M2, or M3 = M1*M2.

TRAN,M1,$\overset{+}{M}$2

This operation generates a new matrix M2 which is the transpose of matrix M1.

SCALE,$\overset{-}{M}$1,M2

This operation replaces each term in the matrix named M1 with the term multiplied by the term M2(1,1) of the matrix named M2.

SOLVE,M1,M2,N1,N2 or SOLVE,M1,M2,,N2 or SOLVE,M1,N1,N2 or
SOLVE,M1,M2,N1 or SOLVE,M1,M2

If N1 = 0, this operation solves the matrix equation AX=B. M1 is the name of the A matrix and M2 is the name of the B matrix. Matrix A is triangularized and the results, X, are stored in M2. If N1=1, Matrix A is triangularized only. If N1=2, for a given B matrix and the A matrix previously triangularized, the B matrix is replaced by the results, X. If N1=3, Matrix A is replaced by its inverse FOR SYMMETRIC MATRICES ONLY.
If N2=0 or blank, matrix A is symmetric. If N2 is nonzero the matrix A is not symmetric.
For symmetric matrices, A is factored into the LDL form. The diagonal D matrix is stored on the diagonal of A. The parameter N2 permits the direct solution of non-symmetric systems of equations. If N2 is not equal to zero, an LU decomposition of matrix A will be performed. No direct replacement of M1 by its inverse is available for the non-symmetric case. Instead, use the ZERO operation to create an identity matrix M2 of the same order as M1. The command SOLVE,M1,M2,,N2 will then replace the matrix M2 with the inverse of the matrix A.

DUPSM,M1,M2,N1,N2,N3,N4

This operation forms a new submatrix named M2 with N3 rows and N4 columns from the terms within the matrix named M1. The first term of matrix M2, M2(1,1), will be from row N1 and column N2 of matrix M1, or M1(N1,N2).

STOSM,M1,M2,N1,N2

This operation stores a submatrix named M2 within the matrix named M1. The first term of the submatrix M2 will be stored at row N1 and column N2 of matrix M1. The terms within the area of M1 in which M2 is stored will be destroyed.

DUPDG,M1,M2

This operation forms a new matrix named M2 from the diagonal terms of M1.

STODG,M1,M2

This operation stores a row or column matrix named M2 at the diagonal locations of matrix M1.

MAX,M1,M2

This operation forms a column matrix named M2 in which each row contains the maximum absolute value of the corresponding row in matrix M1. The maximum and its column number is printed for each row.

NORM,M1,$\overset{+}{\text{M2}}$,N1

If N1 = 0, a row matrix named M2 is formed in which each column contains the sum of the absolute values of the corresponding column of matrix M1. If N1 is not equal to 0, a row matrix named M2 is formed in which each column contains the square root of the sum of the squares of the values of the corresponding columns of matrix M1.

SQREL,$\overset{-}{\text{M1}}$

This operation replaces each term in the matrix named M1 with the square root of the term.

LOG,$\overset{-}{\text{M1}}$

This operation replaces each term in the matrix named M1 with the natural log of the term.

PROD,M1,$\overset{+}{\text{M2}}$

This operation forms a 1 x 2 array named M2 which contains the product of all terms in the matrix named M1. The product, X, is stored as two numbers of the form:

X = P*10**E

in which M2(1) = P and M2(2) = E, the exponent.

DELETE,$\overset{-}{\text{M1}}$

This operation will cause the elimination from storage of the array named M1.

CMS,N1

This operation allows the user to issue CMS commands while under the control of CAL-NPS. In general a command that reloads the virtual core will not be allowed. Examples are FORTHX, SCRIPT, XEDIT, any language processor, SORT, and other large system modules such as COPYFILE and MOVFILE. N1 is the number of words in the command (1 to 9 are allowed). Note that a parenthesis - "(" or ")" is counted as a word. All words must be left justified.

The user will be prompted for each word. If this operation is used in an FT04F001 FILE, each word must be on a separate line.

2. Static Analysis Operations

The purpose of this series of operations is to form the total stiffness and diagonal (lumped) mass matrices for systems of two- or three-dimensional elements. For three-dimensional analysis there are beam and truss elements available. For two-dimensional analysis, there is a frame element, a slope/deflection element for beams, and a 4 to 8 node isoparametric finite element available.

After the creation of an array containing the coordinates of the joints of the system, the specification of displacement boundary conditions, the tabulation of material and section properties, the mass and stiffness matrices are formed for each structural member and placed in sequence on low speed storage along with the global equation numbers which are associated with their stiffness terms. In addition, the member force-displacement transformation matrices are formed and stored on a separate low speed storage file along with the appropriate displacement numbers.

The NODES operation is used to specify or generate the geometry of the system. The operation BOUND specifies which joint displacements exist and assigns internal equation numbers to these displacements. Therefore, each joint may have from zero to six displacement degrees of freedom. Tables of material and section properties for the various members are loaded and printed as standard arrays of information.

A special operation, ADDSF, is used for the direct addition of element stiffnesses to form the total stiffness and diagonal mass matrix of the system. The ADDK operation may be used to add individual elements into the total system matrices. The LOADS operation specifies the concentrated joint loads for all load conditions. After the direct solution for joint displacements due to static or dynamic loads,

the member forces can be evaluated using the MEMFRC operation. The DISPL operation is used to print the displacements in joint number order.

NODES,$\overset{+}{\text{M1}}$,N1 or NODES,$\overset{+}{\text{M1}}$,N1,N2

This operation generates an array (N1,3) named M1 which contains the coordinates for all joints in a structural system. N2 is optional. Data must be entered in free format as follows:

Item	Contents
1	Node number
2	X-coordinate
3	Y-coordinate
4	Z-coordinate

If N2 = 1, there is a joint generation and coordinate system conversion capability. Data must be entered in free format as follows:

Item	Contents
1	Node number
2	X-coordinate
3	Y-coordinate
4	Z-coordinate
5	System type
6	Generation code

System type refers to the system used when inputing the data. All coordinates will be converted to the cartesian system for use by CAL-NPS.

System type	System
1	Cartesian
2	Cylindrical, Z axis longitudinal
3	Cylindrical, Y axis longitudinal
4	Cylindrical, X axis longitudinal
5	Spherical

The input data is the same as above with the following correspondence:

Cartesian	Cylindrical	Spherical
X	r	ρ
Y	θ	Θ
Z	Z	ϕ

If the generation code is not zero, the next card is a generation vector for the self generation of nodes. It is formatted as follows:

Item	Contents
1	Node number increment
2	X increment
3	Y increment
4	Z increment
5	Last node number to be generated.

It is assumed that the increments pertain to the same system of reference as the preceeding card.
This operation must be terminated by a line of alternating zeros and blanks.

BOUND,$\overset{+}{M1}$

This operation specifies the displacements which are nonzero for the structural system of joints specified by the NODES operation. Where:

M1 = Name of boundary condition code array to be generated.

This operation is followed by a series of cards containing the following information in free format:

Item	Contents
1	Node number for the first node in a series of nodes with identical displacement specification.
2	Node number for the last node in the series.
3	X-translation
4	Y-translation
5	Z-translation
6	X-rotation
7	Y-rotation
8	Z-rotation
9	Node number increment used to generate conditions for addtional nodes.

A translation or rotation equals: (a) zero for zero or undefined displacements, or (b) one for nonzero displacements to be evaluated by other operations.

If a node boundary condition is not specified, all displacements at that node are assumed zero. Cards may be supplied in any order. If node boundary conditions are specified more than once, the last definition is used. This sequence of data must be terminated by a card of alternating zeros and blanks.

The selection by the user of which nodes have nonzero displacments requires an understanding of the direct stiffness procedure. Displacement degrees of freedom which have no stiffness associated with the displacement must be considered to be undefined since it is not possible to develop an equilibrium equation for that direction. The total number of nonzero displacements specified will be the size of the total stiffness matrix to be defined by the ADDSF operation.

+
BEAM,M1,M2,M3,M4

This operation calculates the element stiffness, mass and force-displacement transformation matrices for 3-D beam members. These arrays are stored in sequence on low speed storage to be used by other operations where:

> M1 is the name of the beam element group
> M2 is the name of the coordinate array
> M3 is the name of the boundary condition array
> M4 is the name of the array which contains beam properties and has been loaded by the standard matrix LOAD operation

One card for each beam in this group of beam elements must follow this operation. The beam cards are punched in free format, where:

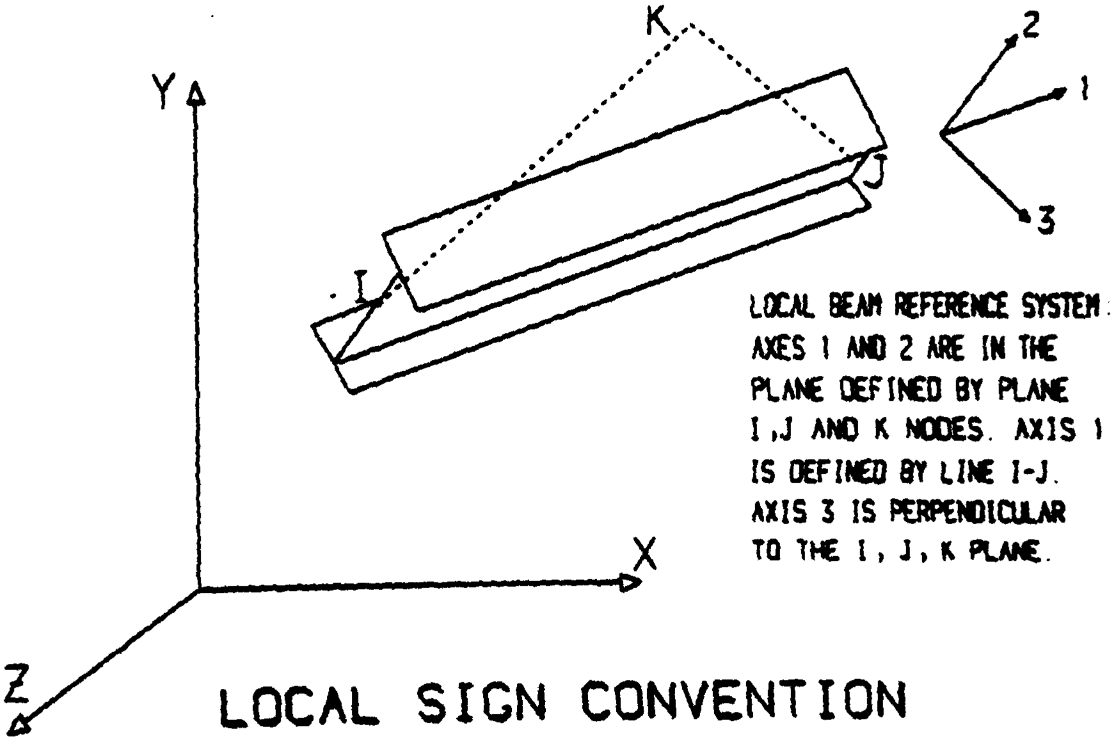
Item	Contents
1	Beam identification number
2	Node number I
3	Node number J
4	Node number K
5	Beam property number NP

This sequence of cards must be terminated with a card of alternating zeros and blanks.

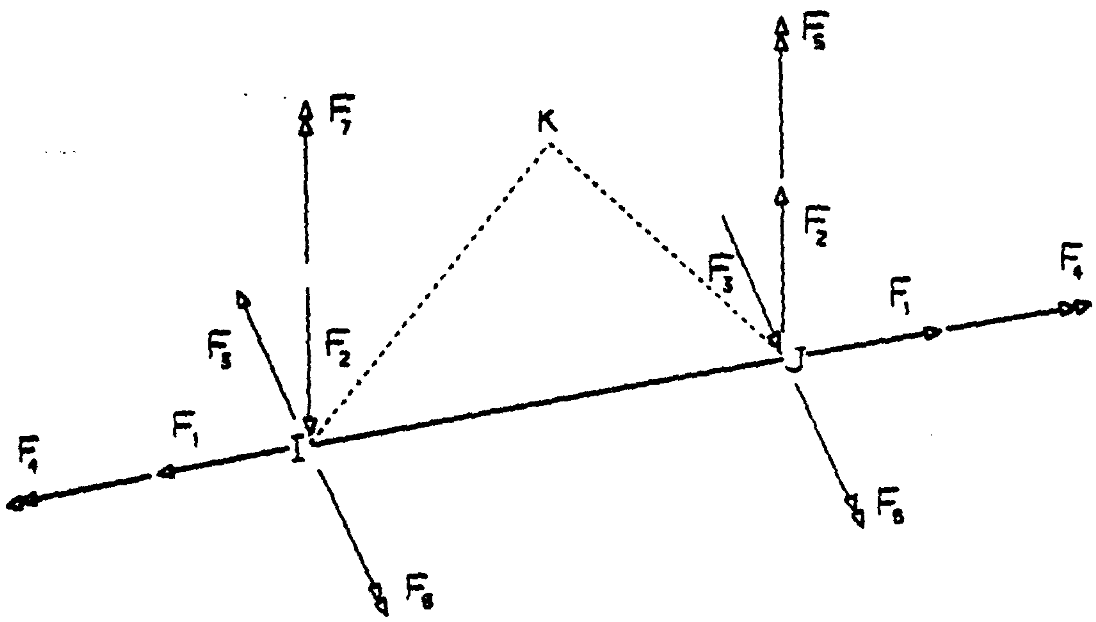
The material and geometric properties for each element are given in the M4 array in the following order:

> M4(NP,1) = Axial area of member, A
> M4(NP,2) = Torsional moment of intertia, J
> M4(NP,3) = Moment of inertia about axis 2, I
> M4(NP,4) = Moment of inertia about axis 3, I
> M4(NP,5) = Modulus of elasticity, E
> M4(NP,6) = Shear modulus, G
> M4(NP,7) = Mass per unit length of beam

where NP is the specific material property number specified in item 5 of the beam card. The local sign convention is given in the following figure.



LOCAL SIGN CONVENTION



DEFINITION OF POSITIVE BEAM FORCES

TRUSS,M1,M2,M3,M4

This operation forms the element stiffness, mass and force-displacement transformation matrices for 3-D truss members. The arrays are stored on low speed storage in sequence and will be used by other structural operations.

M1 is the name of this group of truss members
M2 is the name of the coordinate array
M3 is the name of the boundary condition array
M4 is an NP by 3 array of section properties in which NP is the number of different section properties and

M4(NP,1) = The cross-sectional area, A
M4(NP,2) = The modulus of elasticity, E
M4(NP,3) = the mass per unit length of the member.
This matrix can be loaded by the matrix LOAD operation.

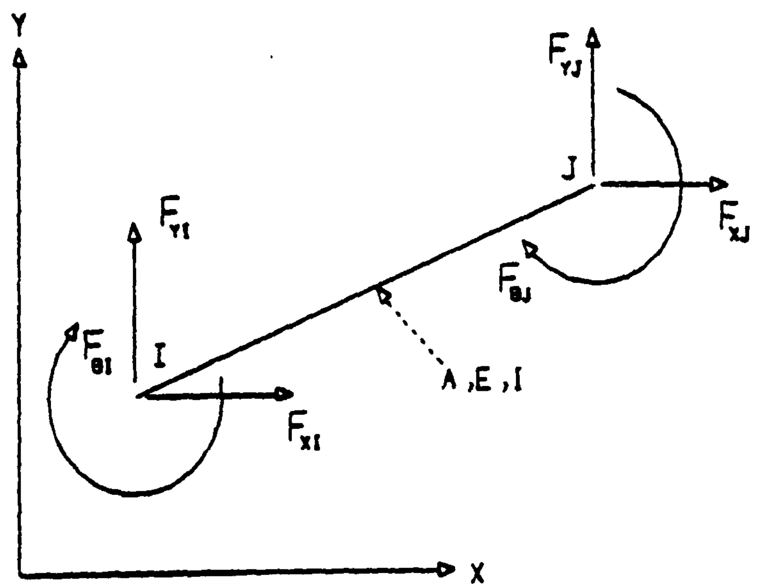
This operation is followed by one card per truss member in free format with the following information:

Item	Contents
1	Truss member identification number
2	Joint number I
3	Joint number J
4	Section property number, NP

This operation must be terminated by a card of alternating zeros and blanks.

FRAME,M1,M2

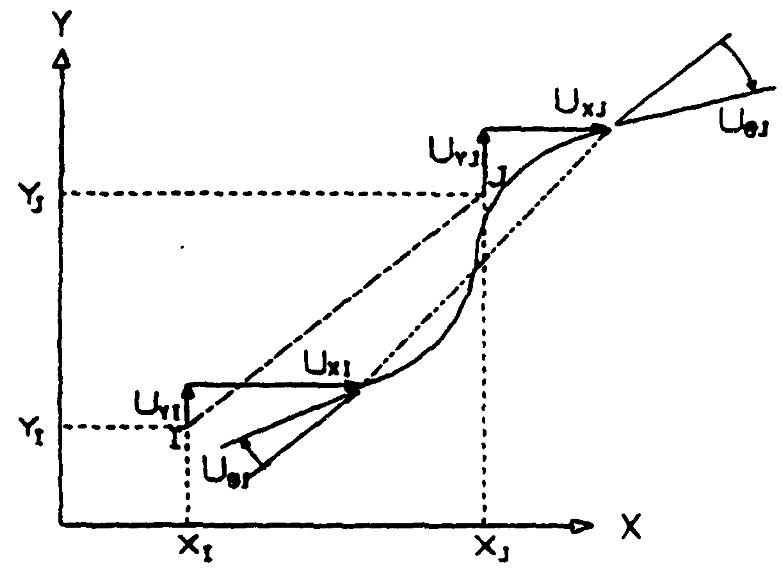
This operation forms the 6 x 6 stiffness matrix for the two-dimensional frame member shown below.



FRAME MEMBER

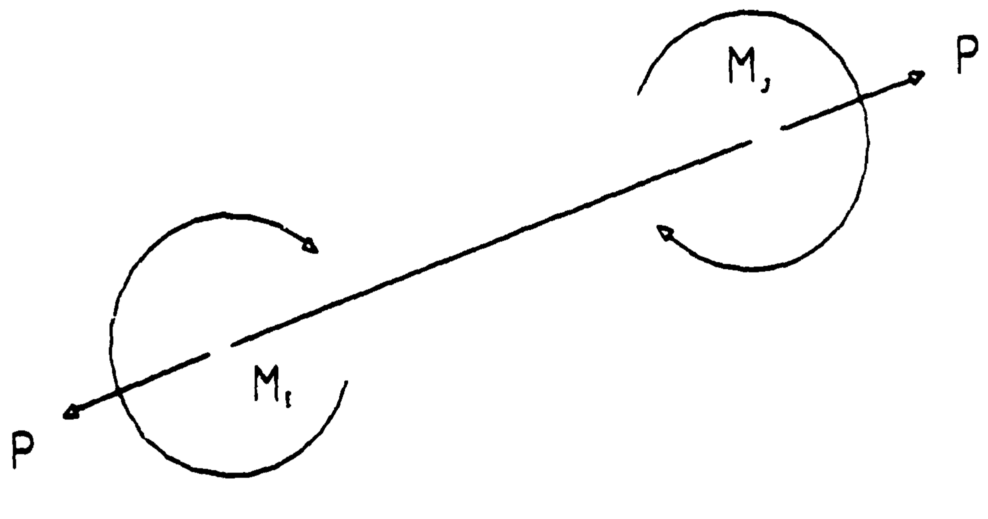
The properties of the member are defined on one cards immediately following the FRAME operation card. This second card is punched in free format and contains the following information:

Item	Contents
1	Axial area, A
2	Modulus of elasticity, E
3	Moment of inertia, I
4	Xi
5	Yi
6	Xj
7	Yj



GEOMETRY AND JOINT DISPLACEMENTS

M2 is a 3 x 6 force-displacement transformation matrix which is based on the positive definition of the element forces shown below.



These forces can be calculated from the following matrix equation with the MEMFRC operation.

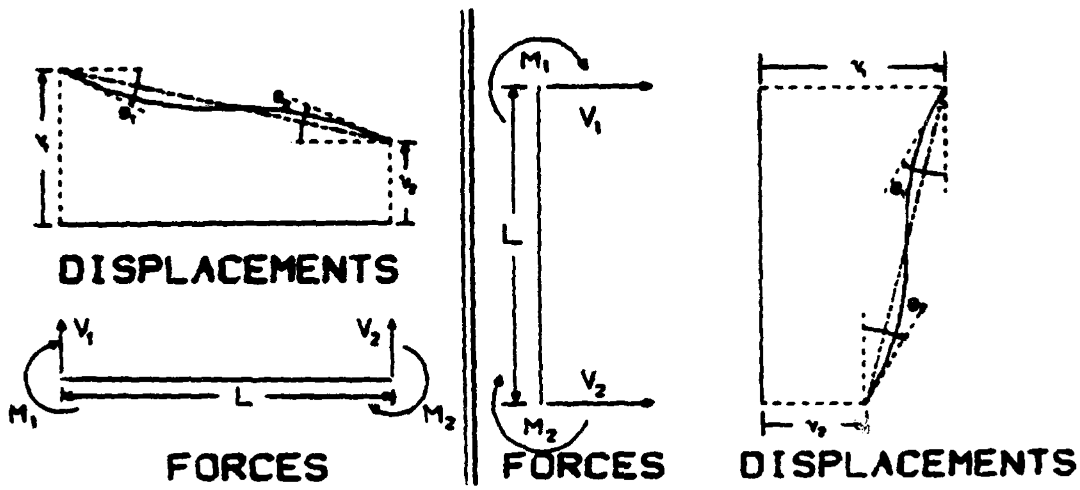
$$\begin{bmatrix} M \\ M \\ P \end{bmatrix} = [M2] \begin{bmatrix} U_{x1} \\ U_{y1} \\ U_{\theta 1} \\ U_{xj} \\ U_{yj} \\ U_{\theta j} \end{bmatrix}$$

SLOPE,M1

This operation forms a 4 x 4 stiffness matrix, M1 for a beam or column member from the classical slope-deflection equations. The properties of the member are defined on one card immediately following the operation. This second card is punched in free format and contains the following information:

Item	Contents
1	Moment of inertia, I
2	Modulus of elasticity, E
3	Length of member, L

The sign convention is defined as follows:



The member forces are defined in terms of joint displacements by the following slope deflection equations.

$$M_1 = \frac{EI}{L}\left[4\theta_1 + 2\theta_2 - \frac{6}{L}(v_1 - v_2)\right]$$

$$M_2 = \frac{EI}{L}\left[2\theta_1 + 4\theta_2 - \frac{6}{L}(v_1 - v_2)\right]$$

$$V_1 = -V_2 = \frac{M_1 + M_2}{L}$$

PLANE,M1,M2,M3,M4,N1,N2

This operation calculates the element stiffness,mass and stress-displacement transformation matrices for 4 to 8 node isoparametric elements (Y-Z plane only). These arrays are stored on low speed storage to be used later by other operations (i.e., ADDSF and FORCE). The arguments are defined as

M1 is the user defined name of the element group
M2 is the name of the joint coordinate array
M3 is the name of the boundary condition array
M4 is the name of the array which contains the material properties of the elements (one row per different material) where

M4 (NP,1) = Modulus of elasticity, E
M4 (NP,2) = Poissons ratio, v
M4 (NP,3) = Thickness of element
M4 (NP,4) = Mass density of the element.

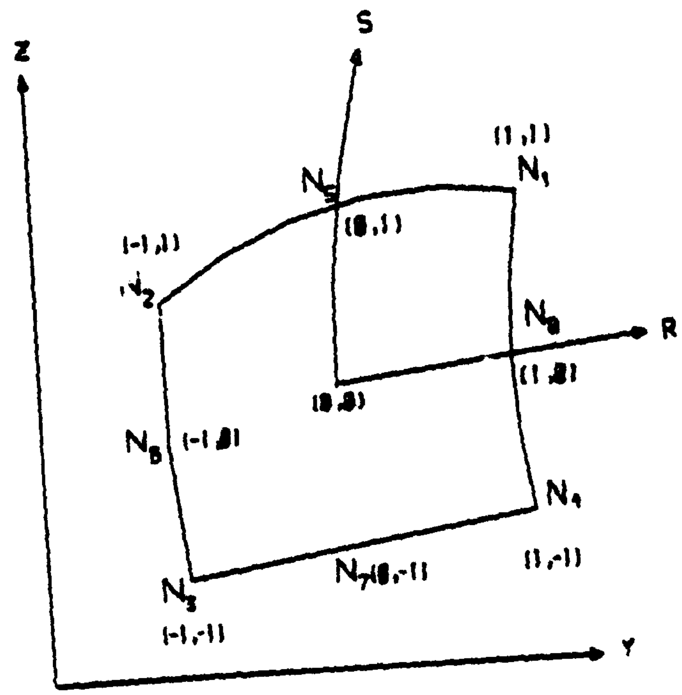
NP is the material identification number.

N1 and N2 are the number of integration points in the r and s directions respectively.

One card for each 3 to 8 node element in the group must follow the operation card. The cards are punched in free format and contain the following information:

Item	Contents
1	Element identification number
2	Node number N1
3	Node number N2
4	Node number N3
5	Node number N4
6	Node number N5
7	Node number N6
8	Node number N7
9	Node number N8
10	Material identification number, NP
11	Natural Coordinate of stress output r1
12	Natural Coordinate of stress output s1
13	Natural Coordinate of stress output r2
14	Natural Coordinate of stress output s2
15	Natural Coordinate of stress output r3
16	Natural Coordinate of stress output s3

N4 through N8 are optional, but zeros must be inserted for them if unused. The midside nodes, if present, must be within the center half of the side. The local numbering system for the element is shown in the following figure.



ISOPARAMETRIC ELEMENT

Stresses will be printed by the FORCE operation at the three points defined in items 11 through 16. The forces are defined as follows:

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \\ F_7 \\ F_8 \\ F_9 \end{bmatrix} = \begin{bmatrix} \sigma_{xx}^1 \\ \sigma_{yy}^1 \\ \tau_{xy}^1 \\ \sigma_{xx}^2 \\ \sigma_{yy}^2 \\ \tau_{xy}^2 \\ \sigma_{xx}^3 \\ \sigma_{yy}^3 \\ \tau_{xy}^3 \end{bmatrix}$$

LOADS,M1,M2,N1

This operation forms a load matrix named M1 of N1 columns (N1 load conditions) where M2 is the name of the boundary condition array generated by the operation BOUND. This operation is followed by a series of cards - one for each loaded joint for each load condition. These cards are punched in free format as follows:

Item	Column
1	Joint number
2	Load condition number
3	Load in X-direction
4	Load in Y-direction
5	Load in Z-direction
6	Moment about X-axis
7	Moment about Y-axis
8	Moment about Z-axis

This series of cards must be terminated by row of alternating zeroes and blanks.

ADDSF,M1 or ADDSF,M1,M2

This operation forms the total stiffness matrix named M1 and the lumped mass matrix named M2 for the structural system from the element stiffness and mass matrices which are stored on low speed storage. These matrices can be printed with the PRINT operation. If M2 is not specified, the row mass matrix M2 will not be formed.

ADDK,M1,M2,M3,N1

This operation adds the element stiffness matrix named M2 to the total stiffness matrix named M1, where M1 was previously defined and initially set to zero. M3 is the name of the integer array in which the column number N1 contains the row or column numbers in the total stiffness matrix where the element stiffness terms are to be added.

DISPL,M1,M2

This operation prints the displacement vector named M1 in joint sequence order, where M2 is the name of the boundary condition array.

MEMFRC,M1,M2,M3,M4,N1

This operation multiplies the element stiffness matrix named M1 by the joint displacement matrix named M2. M3 is the name of the integer array in which the column number N1 contains the row numbers in the displacement matrix, M2, which are to be multiplied by the element stiffness (or force-displacement) matrix, M1. The results of this multiplication are stored in the array named M4.

FORCE,M1,M2,M3 or FORCE,M1,M2

This operation calculates the member forces for a group of elements in which

M1 is the name of the element group
M2 is the name of the displacement matrix
M3 is the name of the matrix in which the forces are stored in the order calculated.

If M3 is not specified, the element forces will be printed only and will not be retained in storage. For the TRUSS element only the member axial force, F, will be calculated for each member. For the BEAM element, eight forces will be printed with reference to the positive definition shown in the BEAM operation.

3. Dynamic Analysis Operations

The following operations were designed to evaluate the dynamic response of structures subjected to arbitrary time-dependent loads. If these operations are used in connection with the standard matrix operations and the structural analysis operations, a dynamic analysis is a relatively simple procedure. The user has the option of using the mode superposition method or a direct step-by-step integration of the dynamic equations of motion. The user may examine the spectra of both input loading and calculated displacements. In addition, the contributions of the individual modes may be evaluated and compared.

The most common and convenient form for time-dependent data to be specified is as straight line segments between given time points. Therefore, an operation which generates values at equal intervals is necessary. Another common characteristic of time-varying loads on structures is that it is normally possible to represent the loads at all points on the structure by the product of two matrices, a column matrix indicating the spatial distribution of loads times a row matrix which indicates the values as a function of various times. If a more complicated loading is required, it is possible to perform more analyses, each within the restrictions of the program, then add the results of each analysis.

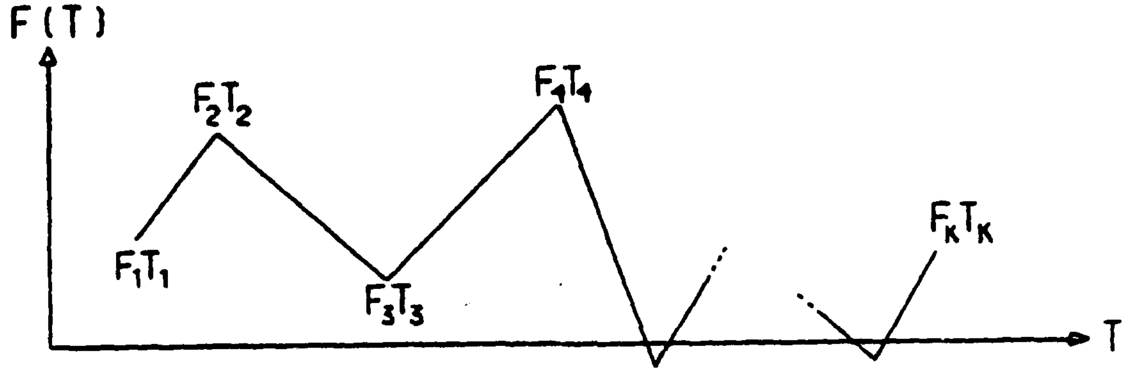
The following operations have been added for the major purpose of performing dynamic analysis.

FUNG,M1,$\overset{+}{M}$2,M3,N1,N2

This operation generates a matrix named M2 which contains values at equal intervals of the function specified in the array named M1. The array M1 must be a 2 by K array of the form:

$$[M1] = \begin{bmatrix} t_1 & t_2 & t_3 & \cdots\cdots & t_k \\ f_1 & f_2 & f_3 & \cdots\cdots & f_k \end{bmatrix}$$

which numerically represents a function of the form shown below:



The time interval t is specified in the 1 by 1 matrix named M3. N1 specifies the total number of values to be generated, and is the number of columns in M2. If N2 = 0, the array M2 will be a 1 x N1 row matrix in which the first value will be f. If N2 is not equal to 0, the array M2 will be a 2 x N1 matrix of the following form:

$$[M2] = \begin{bmatrix} t_1 & t_1 + \Delta t & t_1 + 2\Delta t & \cdots\cdots \\ f_1 & f(t_1 + \Delta t) & f(t_1 + 2\Delta t) & \cdots\cdots \end{bmatrix}$$

STEP,M1,M2,M3,M4,M5,M6,M7,M8,N1,N2

This operation calculates the dynamic response of a structural system using direct step-by-step integration of the following linear matrix equation of motion:

$$[M]\{U\} + [C]\{U\} + [K]\{U\} = R(t) = PF(t)$$

Where:

M1 is the name of the N x N stiffness matrix K
M2 is the name of the N x N mass matrix M
M3 is the name of the N x N damping matrix C
M4 is the name of the N x 3 initial condition matrix U in which:
  U (I,1) is a vector of displacements U
  U (I,2) is a vector of velocities U
  U (I,3) is a vector of accelerations U

M5 is the name of the N by N2 matrix of calculated displacements in which column i represents the displacements at time i*N1* t
M6 is the name of the N x 1 load distribution matrix P
M7 is the name of the 1 x k row matrix representing the load multipliers at equal time increments F, where k = N2/N1
M8 is the name of the 1 x 1 matrix containing t
N1 is the output interval for the displacements
N2 is the total number of displacement vectors to be calculated.

The total time for which results will be calculated by this operation is N1*N2* t. This operation must be followed with one data card in free format containing the following information:

Item	Contents
1	DELTA
2	ALPHA
3	THETA

Different values of delta, alpha and theta will allow the user to select different methods of step-by-step integration. The following table lists some possibilities:

	DELTA	ALPHA	THETA
Newmarks Average Acceleration	1/2	1/4	1.0
Linear Acceleration	1/2	1/6	1.0
Wilson's Theta Method (low damping)	1/2	1/6	1.42
Wilson's Theta Method (high damping)	1/2	1/6	2.0

EIGEN,$\bar{M}1$,$\overset{+}{M}2$,$\bar{M}3$,N1

This operation solves the following eigenvalue problem:

$$K \Phi = M \Phi \lambda$$

In which the N x N, symetric, positive-semidefinite matrix K is named M1. The matrix M is a diagonal matrix of nonzero, positive terms designated by M3. The matrix M3 must be a row or column matrix containing only the diagonal terms of M. The eigenvalues, , are stored in matrix M3. The eigenvalues are ordered in numerically increasing order and the eigenvectors, , are stored in the corresponding columns of the matrix M2. The number N1 specifies the approximate number of significant figures of the eigenvalues. If N1 is zero or blank, 4 figure accuracy will be used. The maximum accuracy possible is 16 figures. The use of more than 12 figure accuracy is not recommended.

The program reduces the problem to standard eigenvalue form by the following transformation

where

$$K^{*} = m^{T} K \ m$$

$$I = m^{T} M \ m$$

in which

$$m_{i} = 1/ \sqrt{M_{ii}}$$

The calculated mode shapes, , are normalized as follows:

$$\Phi^{T} M \Phi = I \qquad \Phi^{T} K \Phi = \lambda$$

The program uses the standard Jacobi diagonalization method to solve for all eigenvalues and eigenvectors.

DYNAM,M1,M2,M3,M4,M5,M6,N1

This operation evaluates the following set of uncoupled second order differential equations associated with the mode superposition method for the dynamic analysis of a structural system.

$$\ddot{x}_i + 2\lambda_i\omega_i\dot{x}_i + \omega_i^2 x_i = P_i(t) \quad i = 1 \text{ to } N \text{ nodes}$$

M1 is the name of a row or column matrix which contains the N terms (frequencies in rad/sec). M2 is the name of a row or column matrix which contains the N terms (ratio of modal damping to critical damping).

The generalized time-varying forces P (t) are not specified directly but are evaluated from more fundamental information. The forces for all modes are evaluated at specific times by the program from the following matrix equation:

p = p*f = M3*M4

In which p is a specified N x 1 vector named M3, and f is a 1 x N1 row matrix which will be generated from the 2 by k array named M4. The array M4 is the same form as the input array described under the operation FUNG. It is not necessary to use FUNG before the DYNAM operation.

M5 is the name of the N x N1 array which contains the generalized displacement X(t).

M6 is the name of the 1 x 1 array which contains the time increment associated with the generalized displacements.

N1 is the number of displacements to be generated.

The method of integration used is exact for straight line segments.

PLOT,M1,N1

This operation will prepare a printer plot of selective rows of the matrix named M1. N1 is the number of rows of M1 which will be plotted by this operation. This operation is followed by N1 cards in f (1a1,I4) format with the following information:

Columns	Contain
1	Plot symbol - any keypunch symbol
2 - 5	Row number to be plotted

The program automatically searches the information to be plotted for the maximum and minimum values. The difference in these numbers divided by 120 spaces is selected as the plot scale.

4. Heat Transfer Operations

The purpose of this series of operations is to form the total conductivity and heat capacity matrices for systems of two- or three-dimensional elements, to form the flux vector and solve the defined set of equations. For two dimensional elements there are 4 to 9 node isoparametric elements. For three dimensional elements there are 8 to 21 node isoparamtric elements.

After the creation of an array containing the coordinates of the system nodes, the specification of element connectivity, the specification of material properties and the specification of constant temperature nodes, the conductivity and capacity matrices and the flux vector are formed. The equations are solved by the appropriate equation solver and the temperatures are printed in node order.

The HTXFR operation initializes the problem. The operation COORD is used to specify or generate nodal coordinates. Element connectivity is specified by the ELCON operation. Material properties and element boundary conditions are input via the PROP operation. The CTEMP operation establishes designated nodes as having constant temperatures. The equation profile for the problem is generated by the PROF operation.

To form the conductivity matrix, the operation SYMC is used. The heat capacity matrix can be approximated with either a consistant (CCAP) or lumped (LCAP) matrix formulation. The flux vector is formed with the FORM operation.

Time independent systems of equations are solved with the CALC operation. Systems of equations involving the first derivative of temperature with respect to time are solved with the ODE series of operations. Nodal temepratures are printed in node number order with the PTEMP operation.

HTXFR

This operation initializes the heat transfer problem. The following information, entered in free format must follow this operation:

Item	Contents
1	Number of nodes (NUMNDP)
2	Number of elements (NUML)
3	Number of material sets (NUMMAT)
4	Spatial dimension (2 or 3) (NDM)
5	Number of unknowns per node (NDF)
6	Maximum number of nodes per element (NEN)

The number of unknowns per node will always be 1 for heat transfer problems.

COORD

This operation creates an array which contains the coordinates for all nodes in a heat transfer system. Data is entered in free format as follows:

Item	Contents
1	Node number
2	X-coordinate
3	Y-coordinate
4	Z-coordinate (if 3-D, else omit)
5	System type
6	Generation code

System type refers to the coordinate system used when inputting data. All coordinates are converted to the cartesian coordinate system for use by CAL-NPS.

System Type	System
0, 1	Cartesian
2	Cylindrical, Z-axis longitudinal
3	Cylindrical, Y-axis longitudinal
4	Cylindrical, X-axis longitudinal
5	Spherical

The input data is the same as shown above with the following correspondance:

Cartesian	Cylindrical	Spherical
X	r	ρ
Y	θ	θ
Z	Z	ϕ

If the generation code is not zero (0), the next card is a generation vector for automatic node generation. It is in free format as follows:

Item	Contents
1	Node number increment
2	X increment
3	Y increment
4	Z increment (if 3-D, else omit)
5	Last node number to be generated

This operation must be terminated by a card with alternating zeroes and blanks.

ELCON

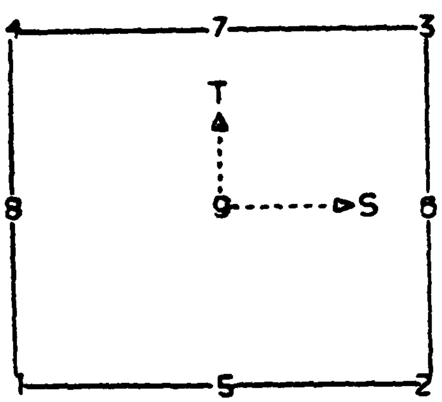
This operation creates an array which contains the element connectivity data for the elements in a heat transfer system. Data is entered in free format as follows:

Item	Contents
1	Element number
2	Node 1 number
3	Node 2 number
etc	etc
N+1	Node N number
N+2	Material set number
N+3	Generation code

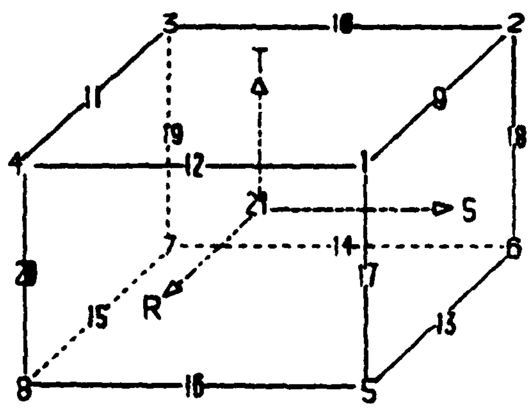
If the generation code is not zero (0), the next card is a generation vector of the automatic generation of element connectivity. It is entered in free format as follows:

Item	Contents
1	Element number increment
2	Node 1 increment
3	Node 2 increment
etc	etc
N+1	Node N increment
N+2	Material set increment (usually 0)
N+3	Last element number to be generated

This operation must be terminated by a row of alternating zeroes and blanks. The node numbering conventions for element connectivity are shown below.



2-D Element



3-D Element

PROP

This operation inputs the material property data for a heat transfer system. The following information, entered in free format, must follow this operation:

Item	Contents
1	Material set number
2	Element type number (2 for 2-D, 3 for 3-D)

Additional information must be provided, depending on the type of element being used. Note that material property data must be in <u>consistent</u> <u>units</u> as shown below:

	<u>English Units</u>	<u>SI Units</u>
k	BTU/hr-ft-°F	W/m-°C
c	BTU/lbm-°F	kJ/kg-°C
rho	lbm/ft³	kg/m³
q'''	BTU/hr-ft³	W/m³

<u>2-D Elements</u>

This information must follow the PROP operation card and its two required entries. An entry must be made for each item. If the item is temperature dependent, the entry will be ignored.

Item	Contents
1	Conductivity in the X-direction
2	Conductivity in the Y-direction
3	Specific Heat
4	Density
5	Heat generation per unit volume
6	Number of integration points per direction (1 to 6, default is 4)
7	Geometry type (see below)
8	Total number of lines with specified boundary conditions in elements with the same material set numer (NLBC) (see below)
9	Temperature dependence code (see below)

Geometry type is 1 for plane geometry and 2 for axisymmetry.

The temperature dependence code is 0 if all material properties are constant and 1 if any property is temperature dependent. If the code is 1, the following information (in free format) is required.

Item	Contents
1	Conductivity in the X-direction code
2	Conductivity in the Y-direction code
3	Heat capacity (specific heat*density) code
4	Heat generated per unit volume code

where 0 means a constant property and 1 means a temperature dependent property. Temperature dependent properties are entered in the form of a table. The tables are consecutively input for conductivity in the X-direction, conductivity in the Y-direction, heat capacity and heat generated per unit volume. Omit the tables for which the temperature dependence code is zero. Tables are input in free format as shown on the following page.

Item	Contents
1	Number of data pairs to be entered (This should be a single card)
2	Temperature 1
3	Property 1 (These two entries should be on one card)
4	Temperature 2
5	Property 2 (These two entries should be on one card)
etc	etc
2*N	Temperature N
2*N+1	Property N (These two entries should be on one card)

If any lines have a specified boundary condition (NLBC>0), a card must be submitted for each line. If the same line is subjected to more than one boundary condition, a card must be used for each one of these conditions. The total number of cards must equal NLBC. The information is entered in free format as follows:

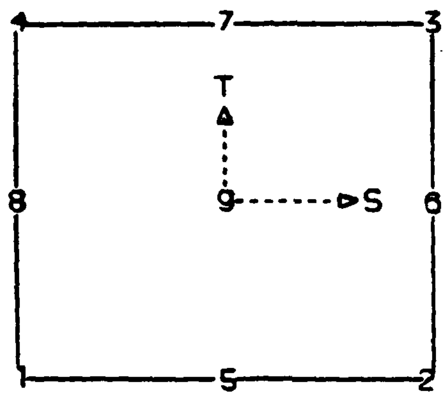
Item	Contents
1	Element number
2	Boundary condition code (see below)
3	Line code (see below)
4	Property value (see below)
5	Ambient temperature

The boundary condition codes are:

1 Flux
2 Convection (constant coefficient)
3 Radiation
4 Convection (temperature dependent property)

The line codes are:

1 S = +1 line
-1 S = -1 line
2 T = +1 line
-2 T = -1 line



The property values are:
Flux - Flux per unit area
Convection - constant heat transfer coefficient (ignored if temperature dependent)
Radiation - Product of emissivity by Stephan-Boltzman constant

The ambient temperature is ignored for the flux boundary condition.
If the boundary condition code is 4 (temperature dependent heat transfer coefficient), a table must follow for the temperature dependence. It is input in free format as is shown on the following page.

Item	Contents
1	Number of data pairs to be entered (This should be a single card)
2	Temperature 1
3	Heat transfer coeficient 1 (These two entries should be on one card)
4	Temperature 2
5	Heat transfer coeficient 2 (These two entries should be on one card)
etc	etc
2*N	Temperature N
2*N+1	Heat transfer coeficient N (These two entries should be on one card)

3-D Elements

This information must follow the PROP operation card and its two required entries. An entry must be made for each item. If the item is temperature dependent, the entry will be ignored.

Item	Contents
1	Conductivity in the X-direction
2	Conductivity in the Y-direction
3	Conductivity in the Z-direction
4	Specific Heat
5	Density
6	Heat generation per unit volume
7	Number of integration points per direction (1 to 6, default is 4)
8	Geometry type (see below)
9	Total number of surfaces with specified boundary conditions in elements with the same material set numer (NSBC) (see below)
10	Temperature dependence code (see below)

Geometry type is 1 for plane geometry and 2 for axisymmetry.

The temperature dependence code is 0 if all material properties are constant and 1 if any property is temperature dependent. If the code is 1, the following information (in free format) is required.

Item	Contents
1	Conductivity in the X-direction code
2	Conductivity in the Y-direction code
3	Conductivity in the Z-direction code
4	Heat capacity (specific heat*density) code
5	Heat generated per unit volume code

where 0 means a constant property and 1 means a temperature dependent property. Temperature dependent properties are entered in the form of a table. The tables are consecutively input for conductivity in the X-direction, conductivity in the Y-direction, conductivity in the Z-direction, heat capacity and heat generated per unit volume. Omit the tables for which the temperature dependence code is zero. Tables are input in free format as is shown on the following page.

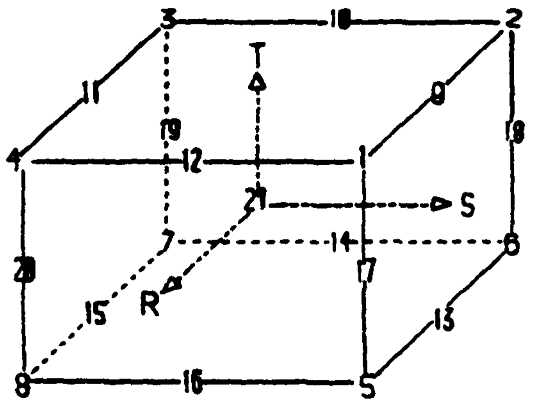
Item Contents
1 Number of data pairs to be entered (This should be a single card)
2 Temperature 1
3 Property 1 (These two entries should be on one card)
4 Temperature 2
5 Property 2 (These two entries should be on one card)
etc etc
2*N Temperature N
2*N+1 Property N (These two entries should be on one card)

If any surfaces have a specified boundary condition (NSBC>0), a card must be submitted for each surface. If the same surface is subjected to more than one boundary condition, a card must be used for each one of these conditions. The total number of cards must equal NSBC. The information is entered in free format as follows:

Item Contents
1 Element number
2 Boundary condition code (see below)
3 Surface code (see below)
4 Property value (see below)
5 Ambient temperature

The boundary condition codes are:
1 Flux
2 Convection (constant coefficient)
3 Radiation
4 Convection (temperature dependent property)

The line codes are:
1 R = +1 line
-1 R = -1 line
2 S = +1 line
-2 S = -1 line
3 T = +1 line
-3 T = -1 line



The property values are:
Flux - Flux per unit area
Convection - constant heat transfer coefficient (ignored if temperature dependent)
Radiation - Product of emissivity by Stephan-Boltzman constant

The ambient temperature is ignored for the flux boundary condition.

If the boundary condition code is 4 (temperature dependent heat transfer coefficient), a table must follow for the temperature dependence. It is input in free format as is shown on the following page.

Item	Contents
1	Number of data pairs to be entered (This should be a single card)
2	Temperature 1
3	Heat transfer coeficient 1 (These two entries should be on one card)
4	Temperature 2
5	Heat transfer coeficient 2 (These two entries should be on one card)
etc	etc
2*N	Temperature N
2*N+1	Heat transfer coeficient N (These two entries should be on one card)

CTEMP

This operation inputs constant temperature boundary restraint data. If you have nodes with constant temperature, you must enter the temperatures for those nodes. For nodes with no temperature restrictions, no entries should be made. Automatic generation capability is built into the operation. Data is entered in free format as follows:

Item	Contents
1	Initial node
2	Last node
3	Node increment
4	Temperature

These entries may be repeated until all the constant temperature nodes are entered. This operation must be terminated by a row of alternating zeroes and blanks.

PROF

This operation establishes the profile of the equations for solution of the heat transfer problem. After the issuance of this command, the problem is set and you may not change the node numbers with restrained boundary conditions (constant temperatures). The values of the restrained temperatures may be changed.

SYMC

This operation forms the symmetric conductance matrix for heat transfer problems.

USYMC

This operation forms an unsymmetric conductance matrix.

LCAP

This operation forms a lumped capacitance approximation matrix for heat transfer problems.

CCAP

This operation forms a consistent capacitance approximation matrix for heat transfer problems.

FORM

This operation forms the flux vector for heat transfer problems.

CALC

This operation solves time independent heat transfer problems for temperature and updates the temperature matrix.

ODE,M1

This operation solves the first order ordinary differential equations arising from time dependent heat transfer problems. If M1 is "INIT",the initial conditions are established. This operation must be followed by the following data in free format:

Item	Contents
1	Integration paramter theta for two point scheme (default = 2/3)
2	Integration parameter gamma for three point scheme (default = 1.5)
3	Integration parameter beta for three point scheme (default = 0.8)
4	Maximum temperature difference for time step adjustment
5	Minimum temperature difference for time step adjustment

Default values are obtained by entering zeros for items 1, 2 and 3.

If M1 is "LINE", the two point integration scheme is used.
If M1 is "QUAD", the three point integration scheme is used.

Some suggested values for theta for the two-point scheme as given in Reference 4 are:

	THETA
Crank-Nicolson	1/2
Zienkiewicz	2/3
Bettencourt	3/4
Liniger	.878

Some suggested values for beta and gamma for the three point scheme as given in Reference 4 are:

	Beta	Gamma
Lees	1/3	1/2
Hogge	3/4	1.0
Wood	.646	1.2184
Zienkiewicz	4/5	3/2
Bettencourt	9/10	3/2

PTEMP

This operation prints the nodal temperatures of a heat transfer system in node number order.

TOL,M1

This operation sets the solution convergence tolerance to the value found in the 1x1 matrix named M1.

CONV

This operation performs a temperature convergence test on a heat transfer system. If this operation is used inside a loop (LOOP operation) and the test shows convergence, looping will be terminated.

DTIM,M1

This operation sets the time increment for integration in a heat transfer system to the value found in the 1x1 matrix M1.

ADTIM

This operation advances the time in a heat transfer problem by one time step. The time step is input with the DTIM operation.

EIGV

This operation computes the dominant eigenvalue and eigenvector of the current heat transfer conductance matrix.

PROMPT

This operation permits the prompts for user input to be suppressed without loss of other output. A second PROMPT will restore the prompts unless the print output is suppressed by the NO command.

---

5. Graphics Operations

CAL-NPS has a limited graphics capability for users with PLOT-10 compatible terminals or IBM 3277 Dual Screen terminals. Two- or three-dimensional meshes may be viewed in the X-Y, Y-Z or X-Z planes.

The graphics operations are initialized with the GRAPH operation by which the user specifies which graphics capable terminal is being used. The PLHX operation plots heat transfer meshes and the PLST operation plots structural meshes. The user may title the plots using the TITLE operation.

GRAPH

This operation initializes the graphics package. Presently there are two terminals users may utilize for the graphics operations. A yes must be input immediately following this card to signify that one of the two types of terminals is being used. Then the terminal code must be entered in free format:

Terminal Code	Terminal
1	PLOT-10 Compatible Terminal
2	IBM 3277 Dual Screen

PLHX,N1

This operation plots 2-D and 3-D heat transfer analysis meshes. If the mesh is 3-D, the user may specify the plane of view:

N1 = 1 : X-Y plane
N1 = 2 : Y-Z plane
N1 = 3 : X-Z plane

The default is the X-Y plane, hence N1 is not required for 2-D meshes.

PLST,M1,M2,N1

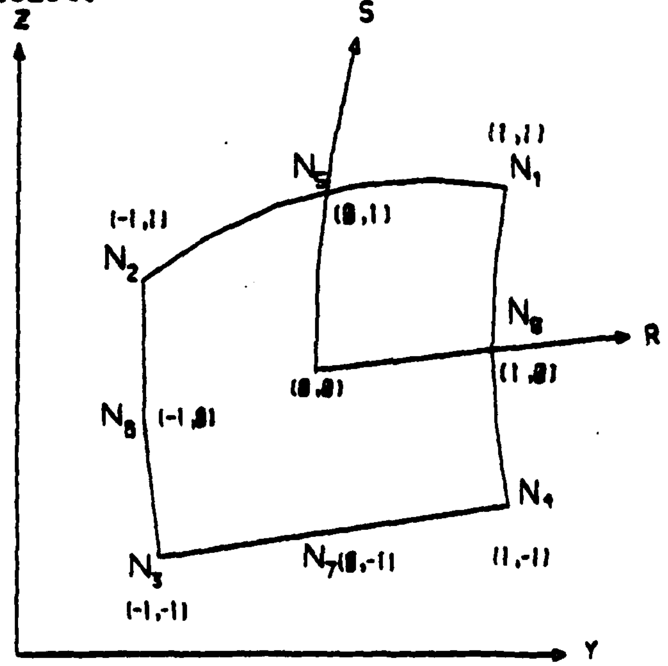
This operation plots 2-D and 3-D structural analysis meshes. M1 is the node coordinate matrix created by the NODES operation. M2 is the element connectivity matrix, loaded with the LOADI operation. The row dimension is the number of elements and the column dimension is the maximum number of nodes per element plus one. M2 contains the following information:

    M2(K,1) = Number of nodes in element number K
    M2(K,2) = Node 1
    M2(K,3) = Node 2
      etc.   etc.
    M2(K,N+1) = Node N

N1 specifies the plane of view:

    N1 = 1 : X-Y plane
    N1 = 2 : Y-Z plane
    N1 = 3 : X-Z plane

If the structure contains two dimensional membrane elements (PLANE operation), the connectivity must follow the convention shown below:



TITLE,N1

This operation allows the user to input N1 (up to three) fifteen character lines to label plots generated with either the PLHX or PLST operations. The label will appear in the upper right hand area of the screen outside the plotting area. This operation must immediately preceed the plotting operation. The title may be changed by reissuing the TITLE command. The default value of N1 is one.

6. Looping Operations

CAL-NPS has a five level looping ability. The first operation is LOOP and the last operation is NEXT. Operations within CAL-NPS are normally executed as they are encountered. If the operation requires data, the data cards follow the operation card. In the case of looping, however, all operation cards are stored within the computer before they are executed. If operations within the loop require data, the data cards must be supplied in the order required after the last NEXT operation. If an error is encountered while executing in a loop, the entire matrix of loop commands is deleted and the user is given the opportunity to try again. Matrices that have been modified by operations successfully completed while in the loop remain modified. After all loops are executed the computer storage required for these operations is automatically released by the program.

LOOP,N1

N1 is the number of times the loop is to be executed. Associated with each LOOP operation there must be a corresponding NEXT operation which signifies the end of the loop and the return of the control to the beginning of the loop. The following is a possible series of looping operations.

```
LOOP,5                                        First level loop
------
------                                        executed 5 times
------
LOOP,2   Second level loop
------   executed twice
------   (total of 10 times)
------
NEXT
------
------
LOOP,4   Second level loop
------   executed 4 times
NEXT     (total 20 times)
NEXT
```

NEXT,M1 or NEXT

The operation NEXT signifies the end of a loop. It is apparent which LOOP and NEXT cards are associated if there are an equal number of each. The operation NEXT,M1 will cause the loop to terminate if the first term in the matrix M1 is negative.

SKIP,M1,N1

This operation will cause the skip of the next N1 operations if the first term in the matrix named M1 is negative.

---

7. <u>User Defined Operations</u>

USERA and USERB

These names are reserved for operations to be defined and programmed by the user. In order to program these operations it is necessary to understand the internal organization of CAL-NPS. Chapter III of Reference 2 contains the details.

E. LARGE PROBLEMS

CAL-NPS is designed as an educational tool. It does not take advantage of banding and symmetry in matrix storage, except in the heat transfer operations. Larger problems can be solved by increasing the dimension of the L array, but a general purpose program that makes maximum advantage of out-of-core storage and takes advantage of banding and symmetry for in-core matrix storage is probably a better choice. With the above disclaimer, to increase problem size capability, increase the dimension of the L array and change the value of MAX to the new dimension size in the following:

```
C-----MAIN PROGRAM
C-----SET PROGRAM CAPACITY
      COMMON MTOT,NDP,L(100000)
      MTOT = 100000
      NDP = 2
      CALL SETIME
      CALL CAL1
      STOP
      END
```

With the dimension of the L array as above, the program currently executes in 1024K bytes for CP/CMS. The region necessary for execution will increase about eight times the increase in the L array.

APPENDIX B

SAMPLE DATA FILE

This is a sample data file (FILE FT04F001) for the hollow cylinder with circumferential heating strips problem.

```
HTXFR
99 80 1
2 1 4
COORD
1 .1666667 5.819103 2 1
11 -.0013019 0 23
2 .1666667 4.252421 2 1
11 -.0013019 0 24
3 .1666667 3.469080 2 1
11 -.0013019 0 25
4 .1666667 2.909551 2 1
11 -.0013019 0 26
5 .1666667 2.461928 2 1
11 -.0013019 0 27
6 .1666667 2.238116 2 1
11 -.0013019 0 28
7 .1666667 2.014305 2 1
11 -.0013019 0 29
8 .1666667 1.790493 2 1
11 -.0013019 0 30
9 .1666667 1.342870 2 1
11 -.0013019 0 31
10 .1666667 0.783341 2 1
11 -.0013019 0 32
11 .1666667 0 2 1
11 -.0013019 0 33
34 .161458 5.819103 2 1
11 -.003906 0 56
35 .161458 4.252421 2 1
11 -.003906 0 57
36 .161458 3.469080 2 1
11 -.003906 0 58
37 .161458 2.909551 2 1
11 -.003906 0 59
38 .161458 2.461928 2 1
11 -.003906 0 60
39 .161458 2.238116 2 1
11 -.003906 0 61
40 .161458 2.014305 2 1
11 -.003906 0 62
41 .161458 1.790493 2 1
11 -.003906 0 63
42 .161458 1.342870 2 1
11 -.003906 0 64
43 .161458 0.783341 2 1
11 -.003906 0 65
44 .161458 0 2 1
11 -.003906 0 66
```

```
67 .147135 5.819103 2 1
11 -.0110675 0 89
68 .147135 4.252421 2 1
11 -.0110675 0 90
69 .147135 3.469080 2 1
11 -.0110675 0 91
70 .147135 2.909551 2 1
11 -.0110675 0 92
71 .147135 2.461928 2 1
11 -.0110675 0 93
72 .147135 2.238116 2 1
11 -.0110675 0 94
73 .147135 2.014305 2 1
11 -.0110675 0 95
74 .147135 1.790493 2 1
11 -.0110675 0 96
75 .147135 1.342870 2 1
11 -.0110675 0 97
76 .147135 0.783341 2 1
11 -.0110675 0 98
77 .147135 0 2 1
11 -.0110675 0 99
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
ELCON
1 12 13 2 1 1 1
1 1 1 1 1 0 10
11 23 24 13 12 1 1
1 1 1 1 1 0 20
21 34 35 24 23 1 1
1 1 1 1 1 0 30
31 45 46 35 34 1 1
1 1 1 1 1 0 40
41 56 57 46 45 1 1
1 1 1 1 1 0 50
51 67 68 57 56 1 1
1 1 1 1 1 0 60
61 78 79 68 67 1 1
1 1 1 1 1 0 70
71 89 90 79 78 1 1
1 1 1 1 1 0 80
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
CTEMP
1 6 1 160
89 99 1 60
0 0 0 0 0 0 0 0 0
PROP
1 2
.1 .1 .6 70 0
4 1 5 0
6 2 2 10 60
7 2 2 10 60
8 2 2 10 60
9 2 2 10 60
10 2 2 10 60
PROP
SYMC
FORM
CALC
PTEMP
READ,5
```

APPENDIX C

EXAMPLE TERMINAL SESSION

The following terminal session was recorded using the data file in Appendix B.

```
CAL
          ENTER TERMINAL CODE:

      1 = PLOT-10 Compatible Terminal (GRAPHICS)
      2 = IBM 3277 DUAL SCREEN (GRAPHICS)
      3 = Any Alpha Numeric Terminal (NO GRAPHICS)

3
IBM 3278 TERMINAL
CP TERMINAL LINESIZE 132
GLOBAL TXTLIB CMSLIB FORTMOD2 MOD2EEH NONIMSL IMSLSP
FILEDEF 01 DISK FILE 01 (RECFM VS LRECL 3408 BLKSIZE 3412)
FILEDEF 08 DISK FILE FT08F001 (RECFM FBA LRECL 132 BLKSIZE
FILEDEF 04 DISK
FILEDEF 13 DISK CAL TEST (RECFM FBA LRECL 132 BLKSIZE 132)
FILEDEF 50 DISK HLP CAL C
LOAD CALO CAL FRTCMX GROUP1 GROUP2 GROUP3 GROUP4 GR (CLEAR
EXECUTION BEGINS...
-----------------------------          0.0   SECONDS
START
**START

-----------------------------          0.003 SECONDS
PROMPT
**PROMPT
-----------------------------          0.003 SECONDS
READ,4
**READ,4
-----------------------------          0.003 SECONDS
**HTXFR
   99 ROWS       1 COLUMNS
    1 ROWS       8 COLUMNS
    2 ROWS       4 COLUMNS
    4 ROWS       1 COLUMNS
    1 ROWS       4 COLUMNS
    4 ROWS       1 COLUMNS
    4 ROWS       4 COLUMNS
    7 ROWS       1 COLUMNS
   10 ROWS       1 COLUMNS
    1 ROWS      99 COLUMNS
    2 ROWS      99 COLUMNS
    5 ROWS      80 COLUMNS
    1 ROWS      99 COLUMNS
   99 ROWS       1 COLUMNS
   99 ROWS       1 COLUMNS
```

```
NUMBER OF NODAL POINTS          =     99
NUMBER OF ELEMENTS              =     80
NUMBER OF MATERIAL SETS         =      1
DIMENSION OF COORDINATE SPACE=         2
DEGREES OF FREEDOM/NODE         =      1
NODES PER ELEMENT (MAXIMUM)     =      4
----------------------------       0.003 SECONDS
```

**COORD

NODE NO	X	Y
1	0.166	0.017
2	0.166	0.012
3	0.166	0.010
4	0.166	0.008
5	0.167	0.007
6	0.167	0.007
7	0.167	0.006
8	0.167	0.005
9	0.167	0.004
10	0.167	0.002
11	0.167	0.0
12	0.165	0.017
13	0.165	0.012
14	0.165	0.010
15	0.165	0.008
16	0.165	0.007
17	0.165	0.006
18	0.165	0.006
19	0.165	0.005
20	0.165	0.004
21	0.165	0.002
22	0.165	0.0
23	0.163	0.017
24	0.164	0.012
25	0.164	0.010
26	0.164	0.008
27	0.164	0.007
28	0.164	0.006
29	0.164	0.006
30	0.164	0.005
31	0.164	0.004
32	0.164	0.002
33	0.164	0.0
34	0.161	0.016
35	0.161	0.012
36	0.161	0.010
37	0.161	0.008
38	0.161	0.007
39	0.161	0.006
40	0.161	0.006
41	0.161	0.005
42	0.161	0.004
43	0.161	0.002
44	0.161	0.0
45	0.157	0.016
46	0.157	0.012
47	0.157	0.010
48	0.157	0.008
49	0.157	0.007
50	0.157	0.006
51	0.157	0.006
52	0.157	0.005
53	0.158	0.004
54	0.158	0.002
55	0.158	0.0
56	0.153	0.016
57	0.153	0.011
58	0.153	0.009

59	0.153	0.008
60	0.154	0.007
61	0.154	0.006
62	0.154	0.005
63	0.154	0.005
64	0.154	0.004
65	0.154	0.002
66	0.154	0.0
67	0.146	0.015
68	0.147	0.011
69	0.147	0.009
70	0.147	0.007
71	0.147	0.006
72	0.147	0.006
73	0.147	0.005
74	0.147	0.005
75	0.147	0.003
76	0.147	0.002
77	0.147	0.0
78	0.135	0.014
79	0.136	0.010
80	0.136	0.008
81	0.136	0.007
82	0.136	0.006
83	0.136	0.005
84	0.136	0.005
85	0.136	0.004
86	0.136	0.003
87	0.136	0.002
88	0.136	0.0
89	0.124	0.013
90	0.125	0.009
91	0.125	0.008
92	0.125	0.006
93	0.125	0.005
94	0.125	0.005
95	0.125	0.004
96	0.125	0.004
97	0.125	0.003
98	0.125	0.002
99	0.125	0.0

-- 0.083 SECONDS

**ELCON

	1	2	3	4	5
1	12	13	2	1	1
2	13	14	3	2	1
3	14	15	4	3	1
4	15	16	5	4	1
5	16	17	6	5	1
6	17	18	7	6	1
7	18	19	8	7	1
8	19	20	9	8	1
9	20	21	10	9	1
10	21	22	11	10	1
11	23	24	13	12	1
12	24	25	14	13	1
13	25	26	15	14	1
14	26	27	16	15	1
15	27	28	17	16	1
16	28	29	18	17	1
17	29	30	19	18	1
18	30	31	20	19	1
19	31	32	21	20	1
20	32	33	22	21	1
21	34	35	24	23	1
22	35	36	25	24	1
23	36	37	26	25	1

24	37	38	27	26	1
25	38	39	28	27	1
26	39	40	29	28	1
27	40	41	30	29	1
28	41	42	31	30	1
29	42	43	32	31	1
30	43	44	33	32	1
31	45	46	35	34	1
32	46	47	36	35	1
33	47	48	37	36	1
34	48	49	38	37	1
35	49	50	39	38	1
36	50	51	40	39	1
37	51	52	41	40	1
38	52	53	42	41	1
39	53	54	43	42	1
40	54	55	44	43	1
41	56	57	46	45	1
42	57	58	47	46	1
43	58	59	48	47	1
44	59	60	49	48	1
45	60	61	50	49	1
46	61	62	51	50	1
47	62	63	52	51	1
48	63	64	53	52	1
49	64	65	54	53	1
50	65	66	55	54	1
51	67	68	57	56	1
52	68	69	58	57	1
53	69	70	59	58	1
54	70	71	60	59	1
55	71	72	61	60	1
56	72	73	62	61	1
57	73	74	63	62	1
58	74	75	64	63	1
59	75	76	65	64	1
60	76	77	66	65	1
61	78	79	68	67	1
62	79	80	69	68	1
63	80	81	70	69	1
64	81	82	71	70	1
65	82	83	72	71	1
66	83	84	73	72	1
67	84	85	74	73	1
68	85	86	75	74	1
69	86	87	76	75	1
70	87	88	77	76	1
71	89	90	79	78	1
72	90	91	80	79	1
73	91	92	81	80	1
74	92	93	82	81	1
75	93	94	83	82	1
76	94	95	84	83	1
77	95	96	85	84	1
78	96	97	86	85	1
79	97	98	87	86	1
80	98	99	88	87	1

0.037 SECONDS

**CTEMP

NODE	TEMPERATURE
1	160.00
2	160.00
3	160.00
4	160.00
5	160.00
6	160.00

```
NODE      TEMPERATURE
  89         60.000
  90         60.000
  91         60.000
  92         60.000
  93         60.000
  94         60.000
  95         60.000
  96         60.000
  97         60.000
  98         60.000
  99         60.000
---------------------------         0.007 SECONDS
**PROP

    MATERIAL SET  1 FOR ELEMENT TYPE 2

         DEGREE OF FREEDOM ASSIGNMENTS     LOCAL      GLOBAL
                                           NUMBER     NUMBER
                                             1          1
    HEAT CONDUCTION ELEMENT

      CONDUCTIVITY      KX =    .1000000          KY =    .1000
      SPECIFIC HEAT            .6000000
      DENSITY                  70.00000
      HEAT GENER/UNIT VOL      .0
        4  GAUSS PTS/DIR
        5  LINES WITH SPECIFIED BOUNDARY CONDITIONS
      PLANE    ANALYSIS
   5 ROWS      3 COLUMNS
   3 ROWS      9 COLUMNS
   5 ROWS      2 COLUMNS

    LINE B.C.

     ELEM    B.C.   LINE  PROPERTY VALUE      TEMPERATURE
        6      2      2     10.00000           60.00000
        7      2      2     10.00000           60.00000
        8      2      2     10.00000           60.00000
        9      2      2     10.00000           60.00000
       10      2      2     10.00000           60.00000
---------------------------         0.027 SECONDS
**PROP
  82 ROWS      1 COLUMNS
---------------------------         0.017 SECONDS
**SYMC
 945 ROWS      1 COLUMNS
---------------------------         0.406 SECONDS
**FORM
    FLUX CONVERGENCE TEST
         RNMAX =      60.927         RN    =      60.927
---------------------------         0.716 SECONDS
**CALC
    ENERGY (DR*A*DR) =   0.224917705 2D+05
---------------------------         0.030 SECONDS
**PTEMP

NODAL TEMPERATURES       TIME    0.0

     NODE         TEMP
        1      0.16000D+03
        2      0.16000D+03
        3      0.16000D+03
        4      0.16000D+03
        5      0.16000D+03
        6      0.16000D+03
        7      0.13970D+03
```

8	0.12995D+03
9	0.11974D+03
10	0.11362D+03
11	0.11094D+03
12	0.15581D+03
13	0.15526D+03
14	0.15419D+03
15	0.15233D+03
16	0.14821D+03
17	0.14317D+03
18	0.13992D+03
19	0.13434D+03
20	0.12568D+03
21	0.11957D+03
22	0.11680D+03
23	0.15163D+03
24	0.15064D+03
25	0.14880D+03
26	0.14597D+03
27	0.14199D+03
28	0.13962D+03
29	0.13664D+03
30	0.13384D+03
31	0.12838D+03
32	0.12349D+03
33	0.12104D+03
34	0.14355D+03
35	0.14214D+03
36	0.13999D+03
37	0.13763D+03
38	0.13530D+03
39	0.13400D+03
40	0.13274D+03
41	0.13142D+03
42	0.12897D+03
43	0.12656D+03
44	0.12520D+03
45	0.13257D+03
46	0.13148D+03
47	0.13016D+03
48	0.12898D+03
49	0.12795D+03
50	0.12744D+03

NODAL TEMPERATURES TIME 0.0

NODE	TEMP
51	0.12693D+03
52	0.12644D+03
53	0.12556D+03
54	0.12473D+03
55	0.12427D+03
56	0.12300D+03
57	0.12244D+03
58	0.12181D+03
59	0.12127D+03
60	0.12084D+03
61	0.12062D+03
62	0.12041D+03
63	0.12021D+03
64	0.11987D+03
65	0.11955D+03
66	0.11937D+03
67	0.10875D+03
68	0.10862D+03
69	0.10848D+03
70	0.10836D+03

```
         71        0.10827D+03
         72        0.10823D+03
         73        0.10819D+03
         74        0.10815D+03
         75        0.10808D+03
         76        0.10801D+03
         77        0.10798D+03
         78        0.85162D+02
         79        0.85163D+02
         80        0.85163D+02
         81        0.85163D+02
         82        0.85163D+02
         83        0.85163D+02
         84        0.85163D+02
         85        0.85163D+02
         86        0.85163D+02
         87        0.85163D+02
         88        0.85163D+02
         89        0.60000D+02
         90        0.60000D+02
         91        0.60000D+02
         92        0.60000D+02
         93        0.60000D+02
         94        0.60000D+02
         95        0.60000D+02
         96        0.60000D+02
         97        0.60000D+02
         98        0.60000D+02
         99        0.60000D+02
---------------------------------        0.043 SECONDS
**READ,5
---------------------------------        0.003 SECONDS
stop
**STOP
R; T=1.80/2.67 12:36:15
```

LIST OF REFERENCES

1. University of California Report No. UC SESM 77-2, CAL Computer Analysis Language for the Static and Dynamic Analysis of Structural Systems, by E.L. Wilson, January, 1977.

2. Elliot, L.B., Non-numerical Applications of Computer Programming in the Construction of Problem Oriented Languages, M.S. and M.E. Thesis, Naval Postgraduate School, Monterey, CA, 1979.

3. Zienkiewicz, O.C., The Finite Element Method, 3rd ed., p. 677-757, McGraw-Hill Book Company (UK) Limited, 1977.

4. Bettencourt, J.M., Finite Element Analysis Program (FEAP) for Conduction Heat Transfer, M.S. and M.E. Thesis, Naval Postgraduate School, Monterey, CA, 1979.

5. Zienkiewicz, O.C., op. cit., p. 568-604.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 69 Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor Gilles Cantin, Code 69Ci Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	10
5.	Professor O.C. Zienkiewicz Civil Engineering Department Singleton Park Swansea SA28PP Wales, UNITED KINGDOM	1
6.	1º Ten. Jorge Bettencourt Direccao Do Serviço De Instrucao Ministerio Da Marinha Lisboa, PORTUGAL	1
7.	Profesor Paul F. Pucci, Code 69Pc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
8.	Professor Matthew D. Kelleher, Code 69Kk Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
9.	Professor David Salinas, Code 69Zc Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
10.	Professor Robert E. Newton, Code 69Ne Department of Mechanical Engineering Naval Postgraduate School Monterey, California 93940	1
11.	LCDR Lael R. Easterling, USN 4243 N.W. 54th St. Oklahoma City, Oklahoma 73112	1
12.	Georges Verchery Department de Genie Mecanique Universite de Technologie 60200 Compiegne, FRANCE	1

13. Jean Louis Armand 1
Institut de Recherche Pour la
Construction Navale
3 Avenue de Grand Champ
78230 le Pecq, FRANCE

14. Professor K. J. Bathe 1
Mechanical Engineering Department
M.I.T.
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

15. William J. Dodge 1
Oak Ridge National Laboratory
Building 9204-1 Box Y
Oak Ridge, Tennessee 36830

16. Professor Edward L. Wilson 1
Structural Engineering Division
Civil Engineering Department
University of California (Berkeley)
Berkeley, California 94720

17. Dr. William J. Stronge (Code 603) 1
Naval Weapons Center
China Lake, California 93955

18. J. E. Serpanos 1
Code 3162
Naval Weapons Center
China Lake, California 93955

19. Dr. Jean Louis Batoz 1
Department de Genie Mecanique
Universite de Technologie
60200 Compiegne, FRANCE

20. Professor Guri Dhatt 1
Centre Technique de l'Informatique
Universite Laval
Quebec, Prov. de Quebec
CANADA GIK 7P4

21. Dr. Gilbert Touzot 1
Centre d'Informatique
Universite de Technologie
60200 Compiegne, FRANCE

22. R. A. Langworthy 1
Applied Technology Laboratories
U.S. Army Research and Technology
Laboratory
Fort Eustis, Virginia 23604

23. E. M. Lenoe 1
Army Materials & Mechanic Research
Center
Arsenal Street
Watertown, Massachusetts 02172

24. Dr. Paris Genalis 1
Naval Ship Research and Development Center
Bethesda, Maryland 20084

25. Code SEC 6734 1
Naval Ship Engineering Center
Philadelphia, Pennsylvania 19112

26. Code 471 1
Department of the Navy
Office of Naval Research
Arlington, Virginia 22217

27. Office of the Secretary of Defense 1
DDR&E
3D1089 Pentagon
Washington, DC 20301

28. R. Rice (Code 6360) 1
Naval Research Laboratory
Washington, DC 20375

29. E. Van Reuth 1
Defense Advanced Research Projects
Agency
1440 Wilson Boulevard
Arlington, Virginia 22209

30. I. Machlin 1
Naval Air System Command
Department of the Navy
Washington, DC 20361

31. B. Probst, MS 49-3 1
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

32. Mr. C. P. Blankenship, MS 105-1 1
NASA-Lewis Research Center
21000 Brookpark Road
Cleveland, Ohio 44135

33. Dr. H. Graham/AFLM/LLM 1
Department of the Air Force
Air Force Materials Laboratory
Wright-Patterson Air Force Base,
Ohio 45433

34. S. Freiman (Code 6363) 1
Naval Research Laboratory
Washington, DC 20375

35. N. M. Geyer/AFML/LLM 1
Air Force Materials Laboratory
Wright-Patterson Air Force Base,
Ohio 45433

36. S. Wiederhorn 1
National Bureau of Standards
Washington, DC 20234

37. LT J. H. Preisel, Jr., USN 1
922 Bernard Road
Peekskill, New York 10566

38. Allan F. Greiner 1
United Technologies Research Center
East Hartford, Connecticut 06108

39. LCDR L. B. Elliot 1
Code 338.14
Long Beach Naval Shipyard
Long Beach, California 90822

40. Martin A Krenzke 1
David W. Taylor Naval Ship
Research and Development Center
A236, Bldg. 19
Carderock Laboratory
Bethesda, Maryland 20084

41. Professor Robert L. Taylor 1
Department of Civil Engineering
University of California
at Berkeley
Berkeley, California 94720

42. J. Gagorik (Code 62 Structures) 1
NAVSEA 03R24
Washington, DC 20362

43. LCDR Warren L. Roberts 1
111 South Lane
Albert Lea, Minnesota 56007

44. Professor Y. S. Shin, Code 69Sg 1
Department of Mechanical Engineering
Naval Postgraduate School
Monterey, California 93940